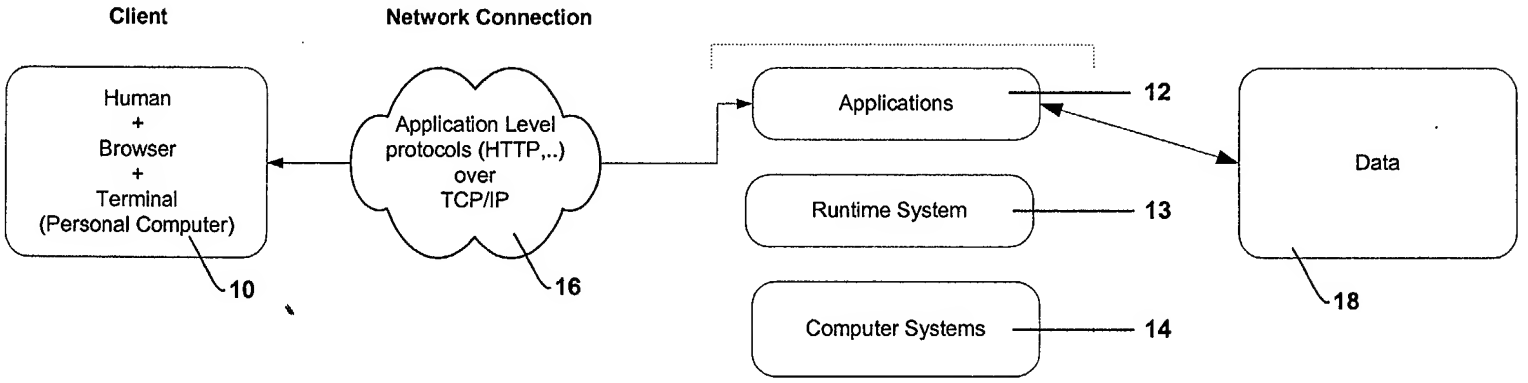

Figure 1 (Prior Art)

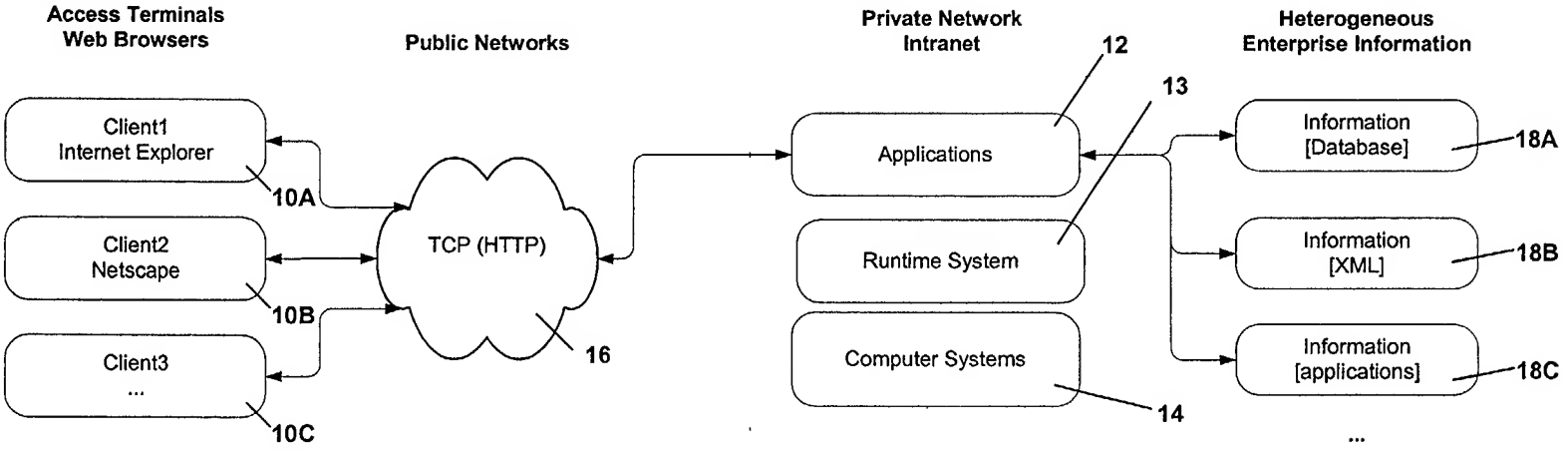

Figure 2 (Prior Art)

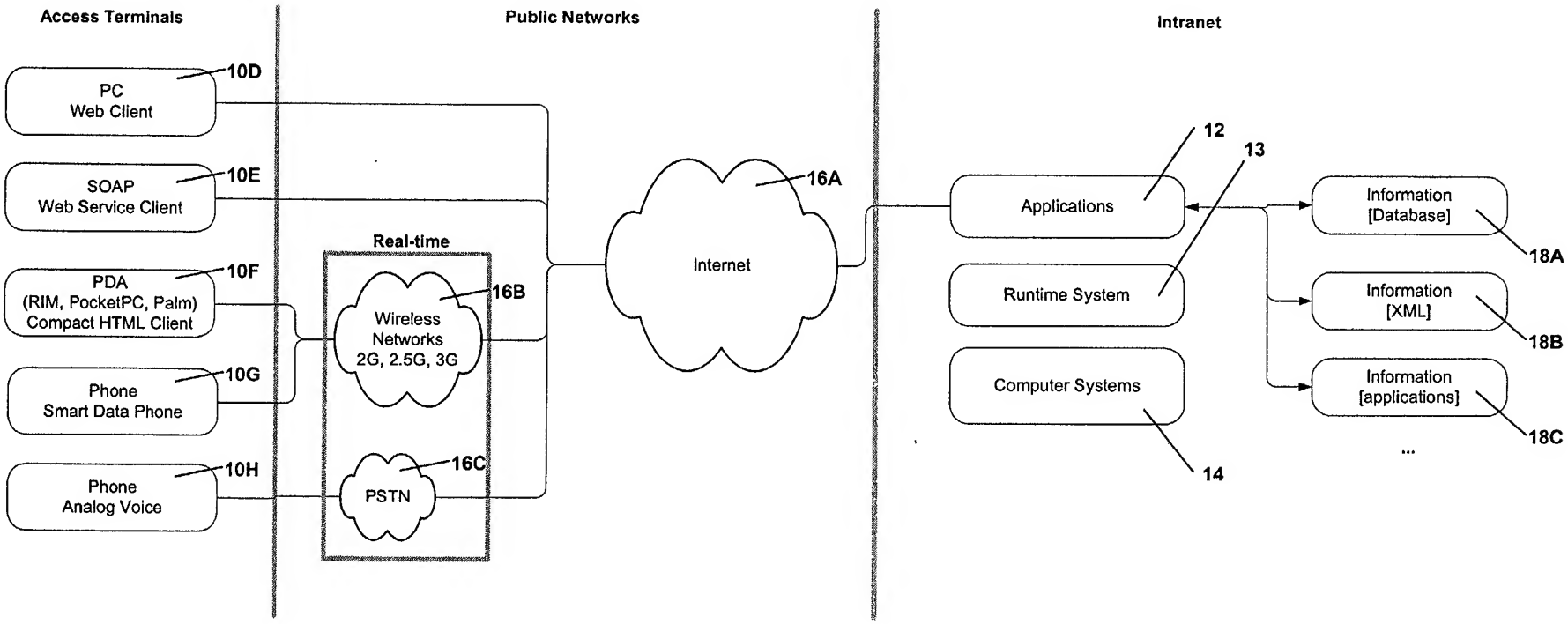

Figure 3 (Prior Art)

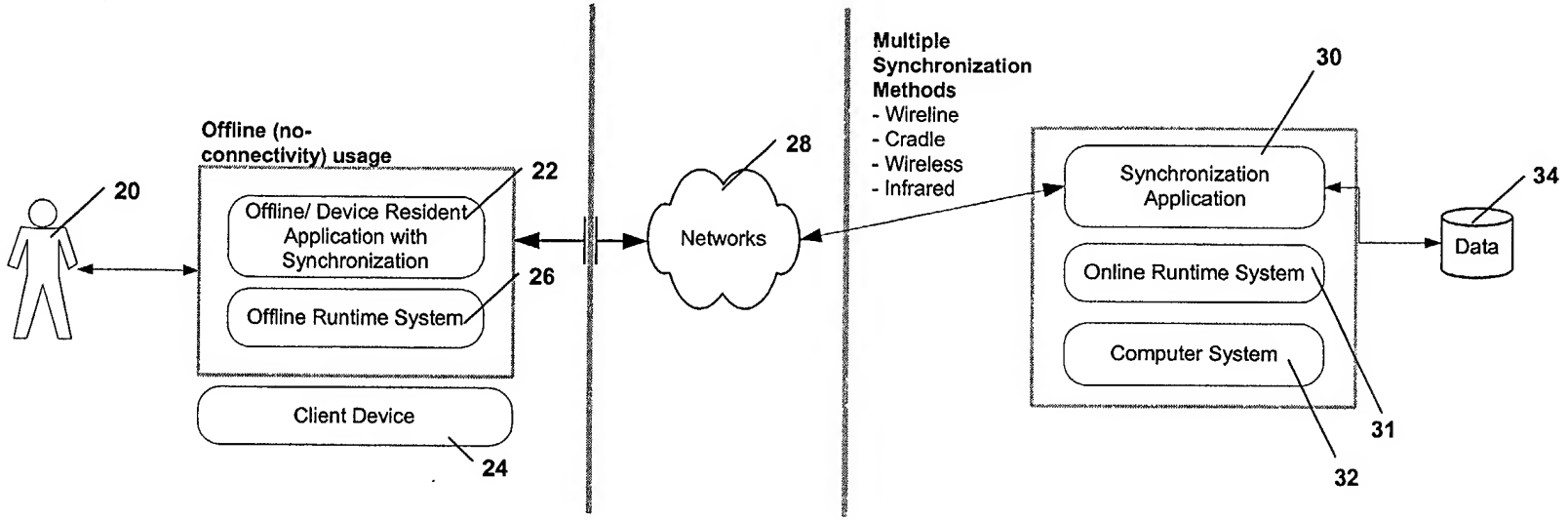

Figure 4 (Prior Art)

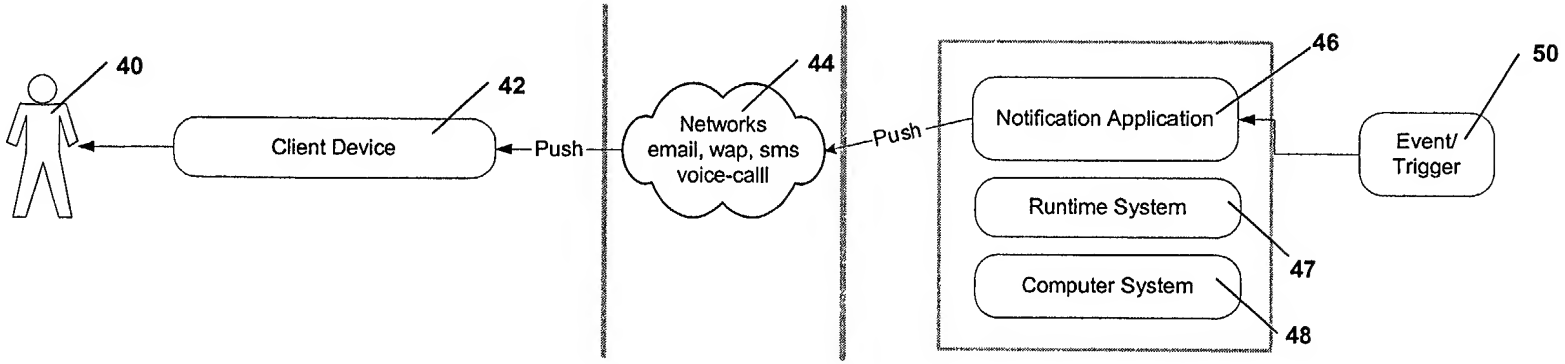

Figure 5 (Prior Art)

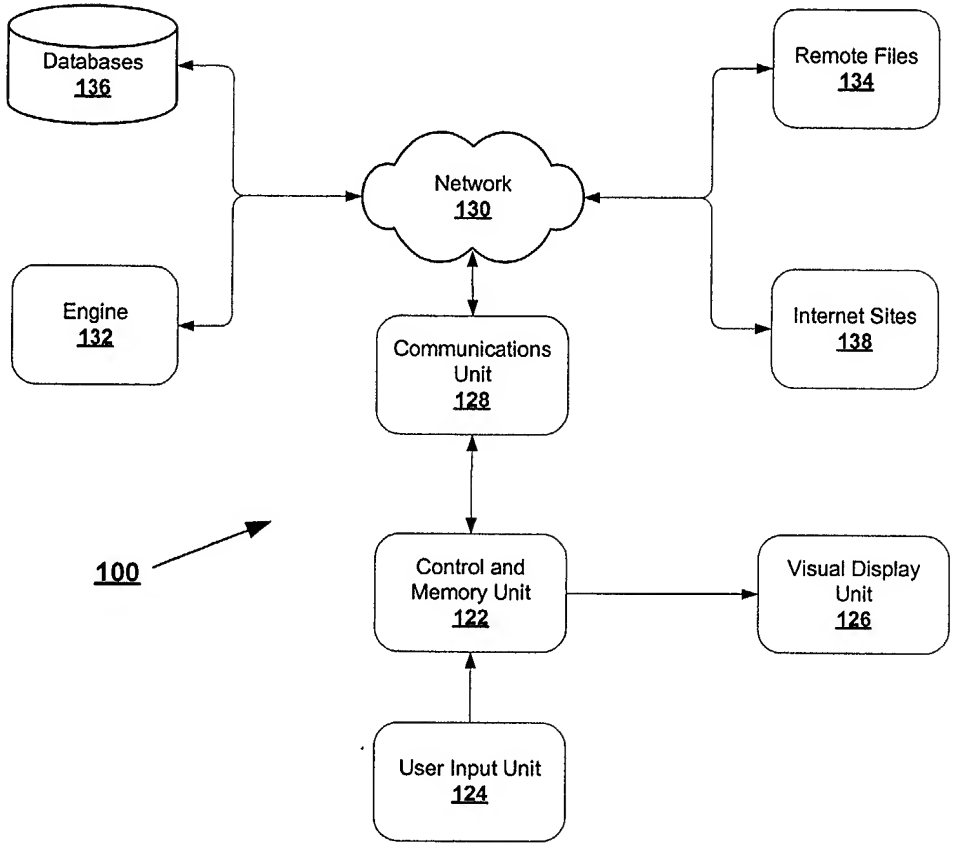
Databases
136
Remote Files
134
Network
130
Engine
132
Internet Sites
138
Communications
Unit
128
100
Control and
Memory Unit
122
Visual Display
Unit
126
User Input Unit
124

Figure 6

Docket Explorer

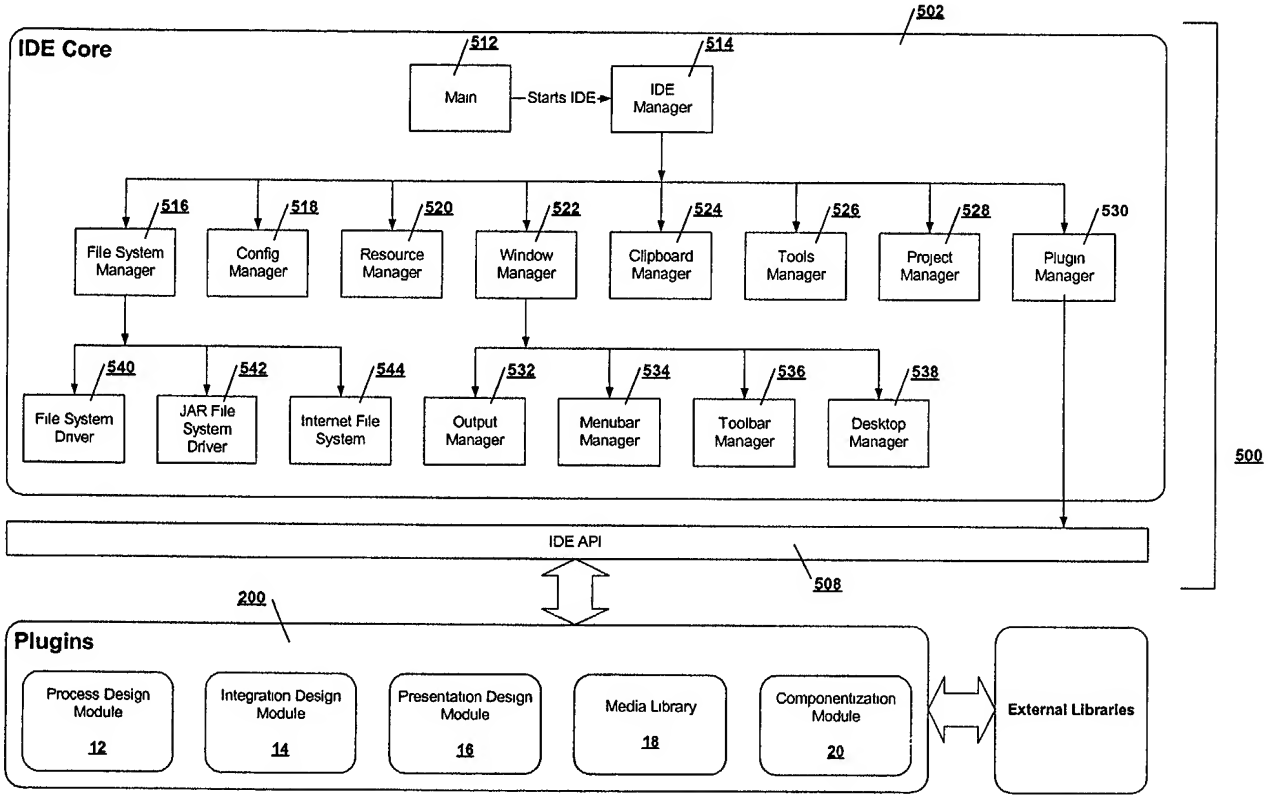

Figure 7

20070619 223 3001

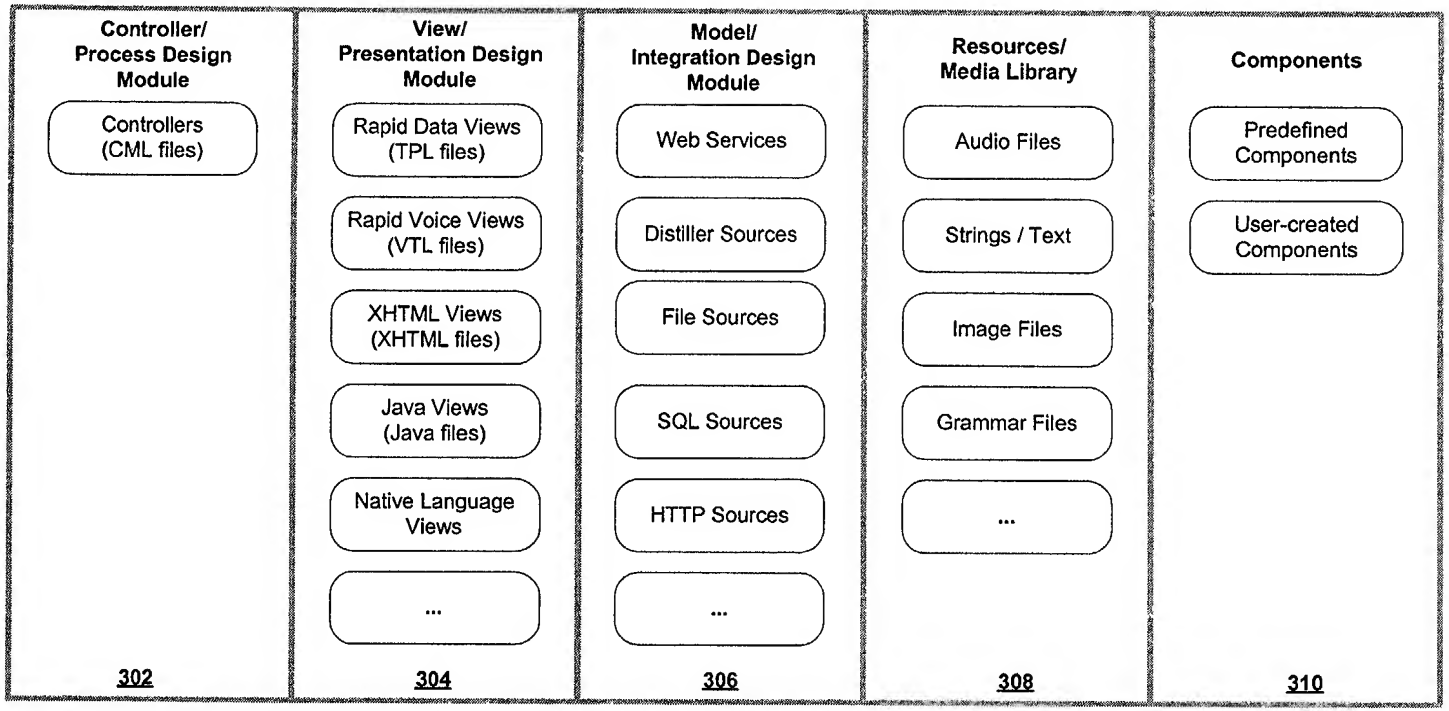

Figure 8

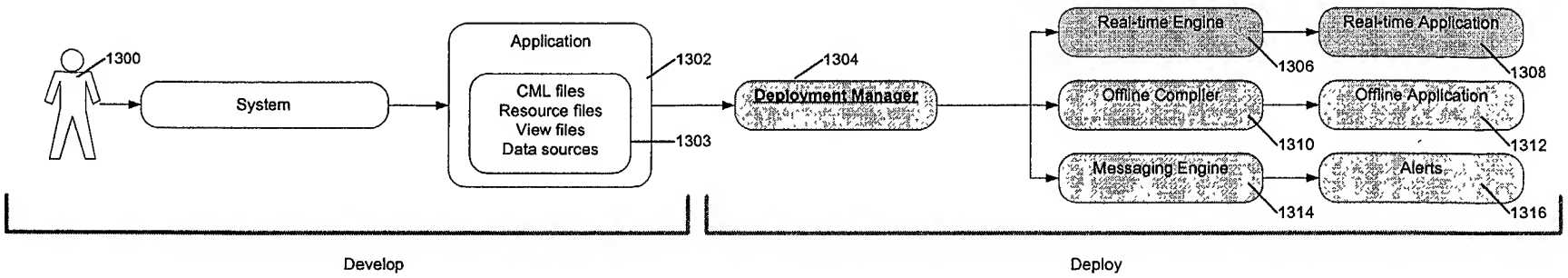

Figure 9

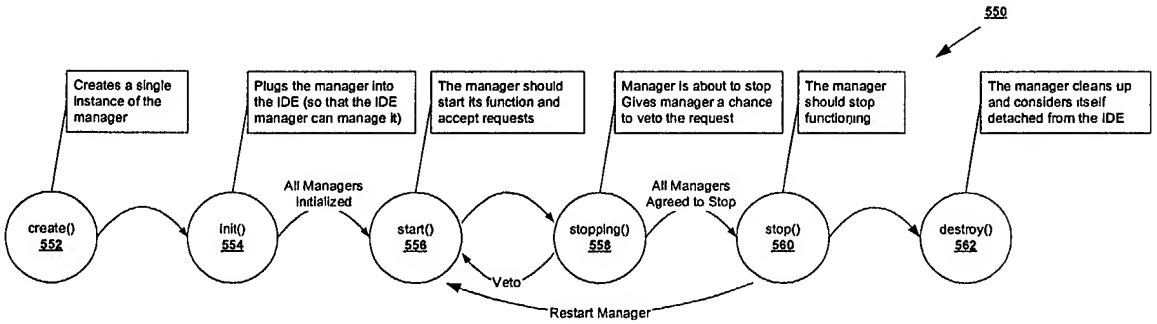

Figure 10

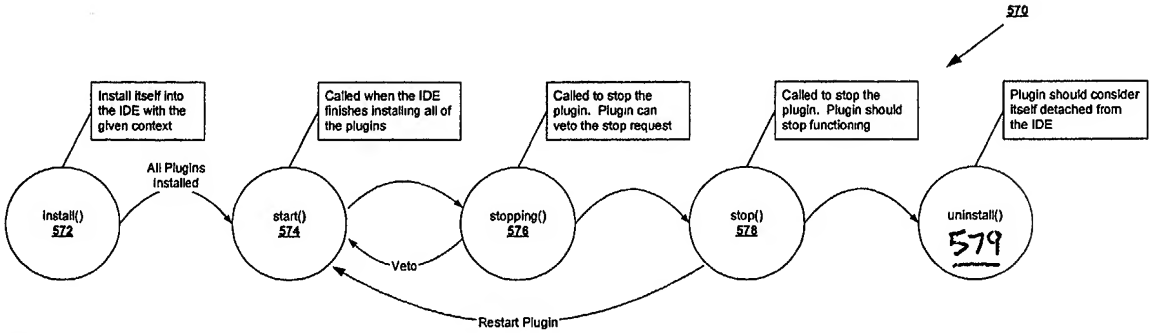
570
Install itself into the IDE with the given context
Called when the IDE finishes installing all of the plugins
Called to stop the plugin. Plugin can veto the stop request
Called to stop the plugin. Plugin should stop functioning
Plugin should consider itself detached from the IDE
All Plugins Installed
install() 572
start() 574
stopping() 576
stop() 578
uninstall() 579
Veto
Restart Plugin

Figure 11

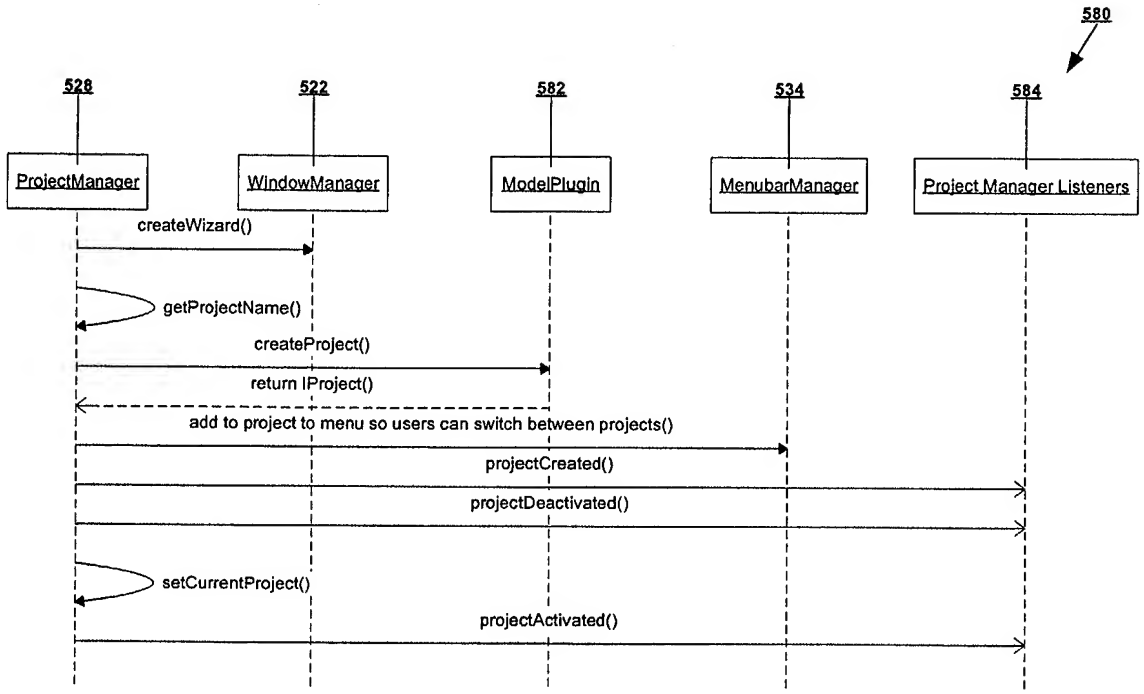
580
528
522
582
534
584
ProjectManager
WindowManager
ModelPlugin
MenubarManager
Project Manager Listeners
createWizard()
getProjectName()
createProject()
return IProject()
add to project to menu so users can switch between projects()
projectCreated()
projectDeactivated()
setCurrentProject()
projectActivated()

Figure 12

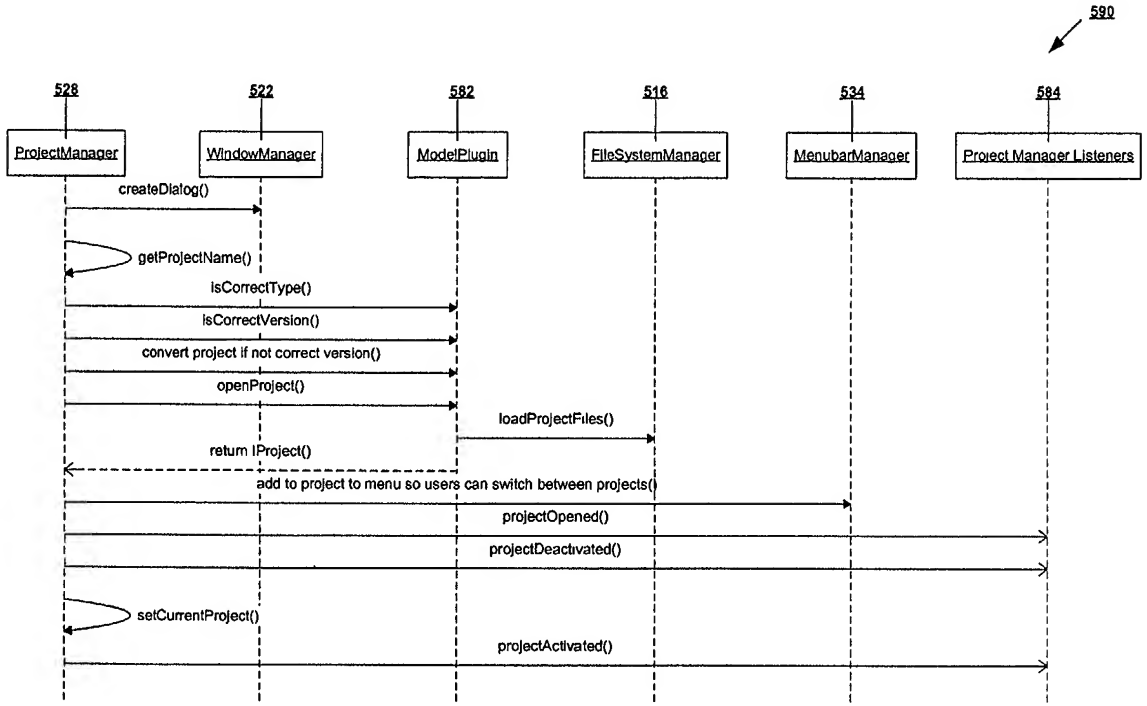
590
528
522
582
516
534
584
ProjectManager
WindowManager
ModelPlugin
FileSystemManager
MenubarManager
Project Manager Listeners
createDialog()
getProjectName()
isCorrectType()
isCorrectVersion()
convert project if not correct version()
openProject()
loadProjectFiles()
return IProject()
add to project to menu so users can switch between projects()
projectOpened()
projectDeactivated()
setCurrentProject()
projectActivated()

Figure 13

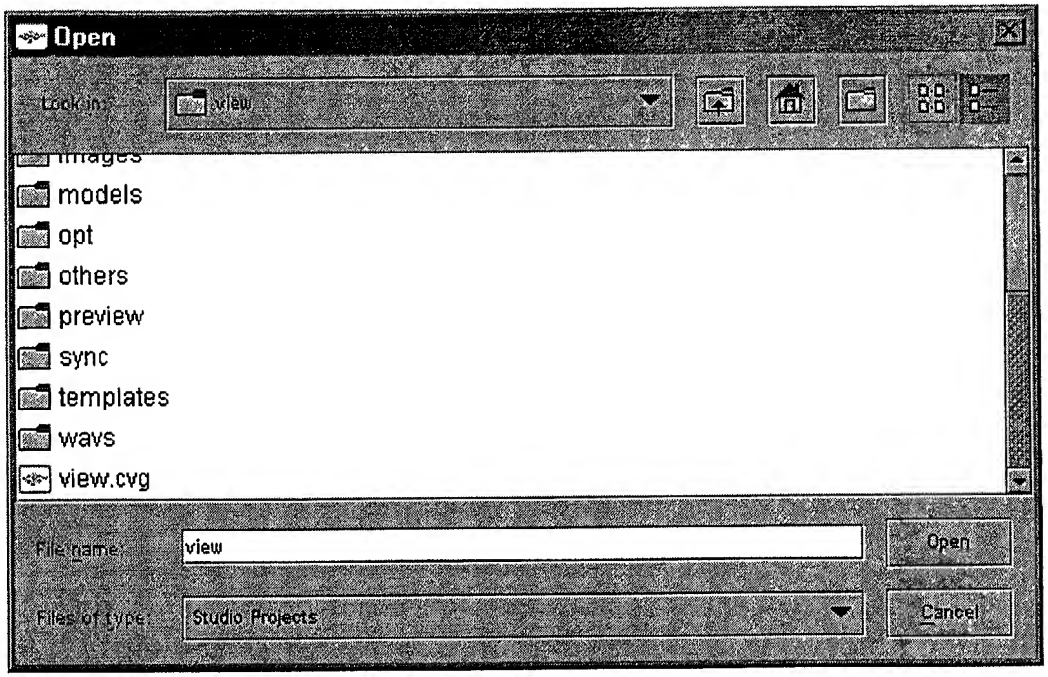
Open
Look in:
view
models
opt
others
preview
sync
templates
wavs
view.cvg
File name:
view
Open
Files of type:
Studio Projects
Cancel

Figure 14

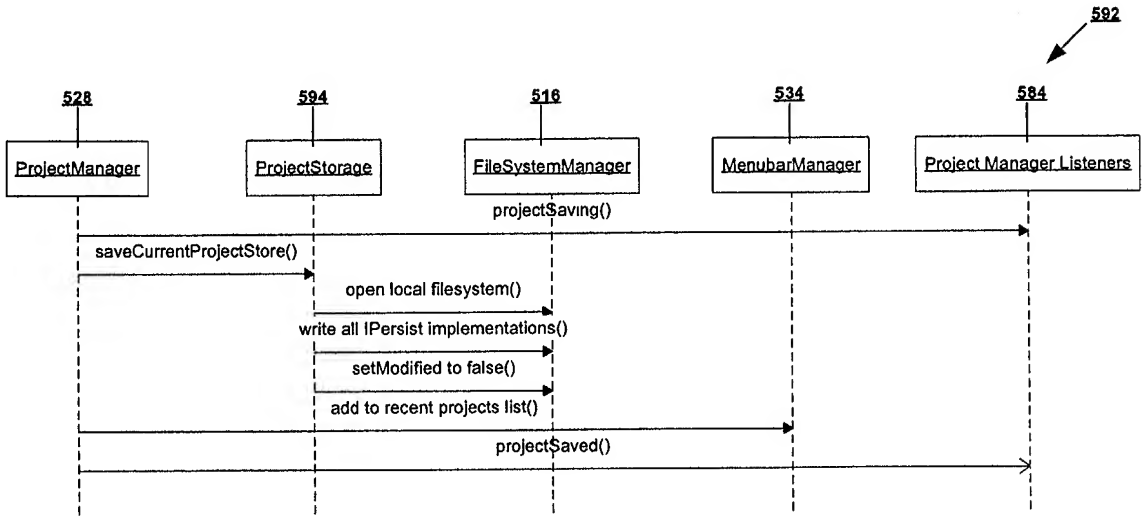
592
528
594
516
534
584
ProjectManager
ProjectStorage
FileSystemManager
MenubarManager
Project Manager Listeners
projectSaving()
saveCurrentProjectStore()
open local filesystem()
write all IPersist implementations()
setModified to false()
add to recent projects list()
projectSaved()

Figure 15

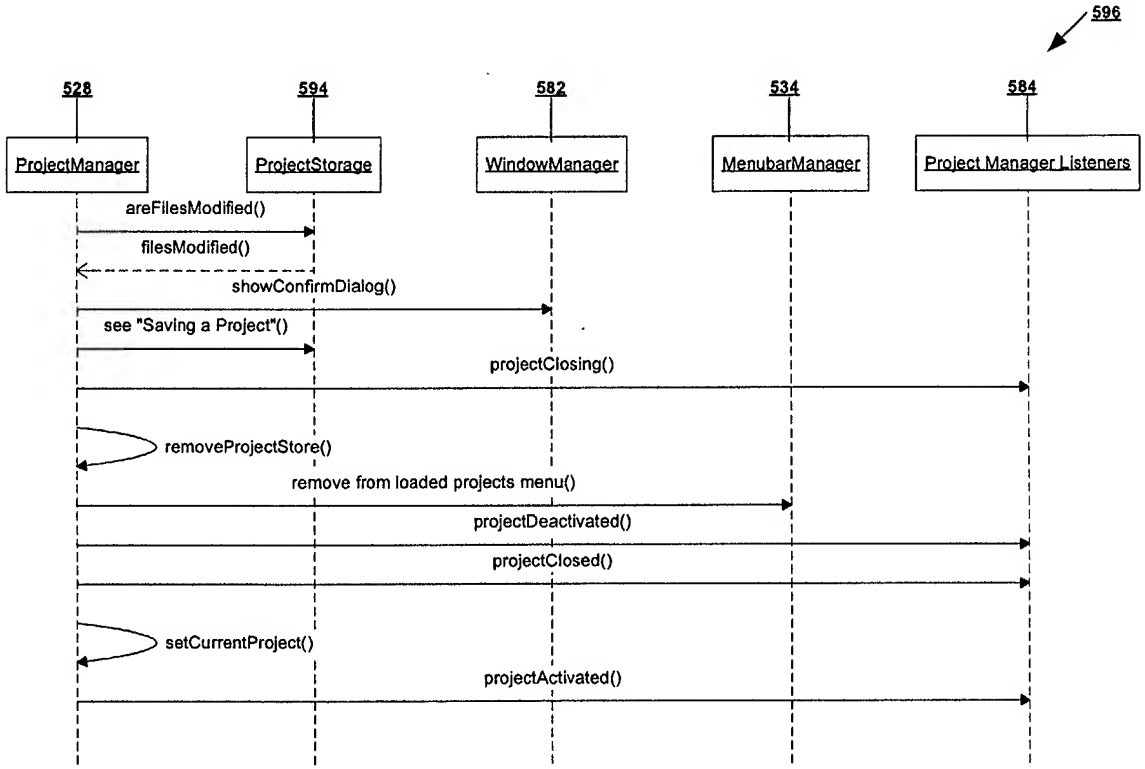
596
528
594
582
534
584
ProjectManager
ProjectStorage
WindowManager
MenubarManager
Project Manager Listeners
areFilesModified()
filesModified()
showConfirmDialog()
see "Saving a Project"()
projectClosing()
removeProjectStore()
remove from loaded projects menu()
projectDeactivated()
projectClosed()
setCurrentProject()
projectActivated()

Figure 16

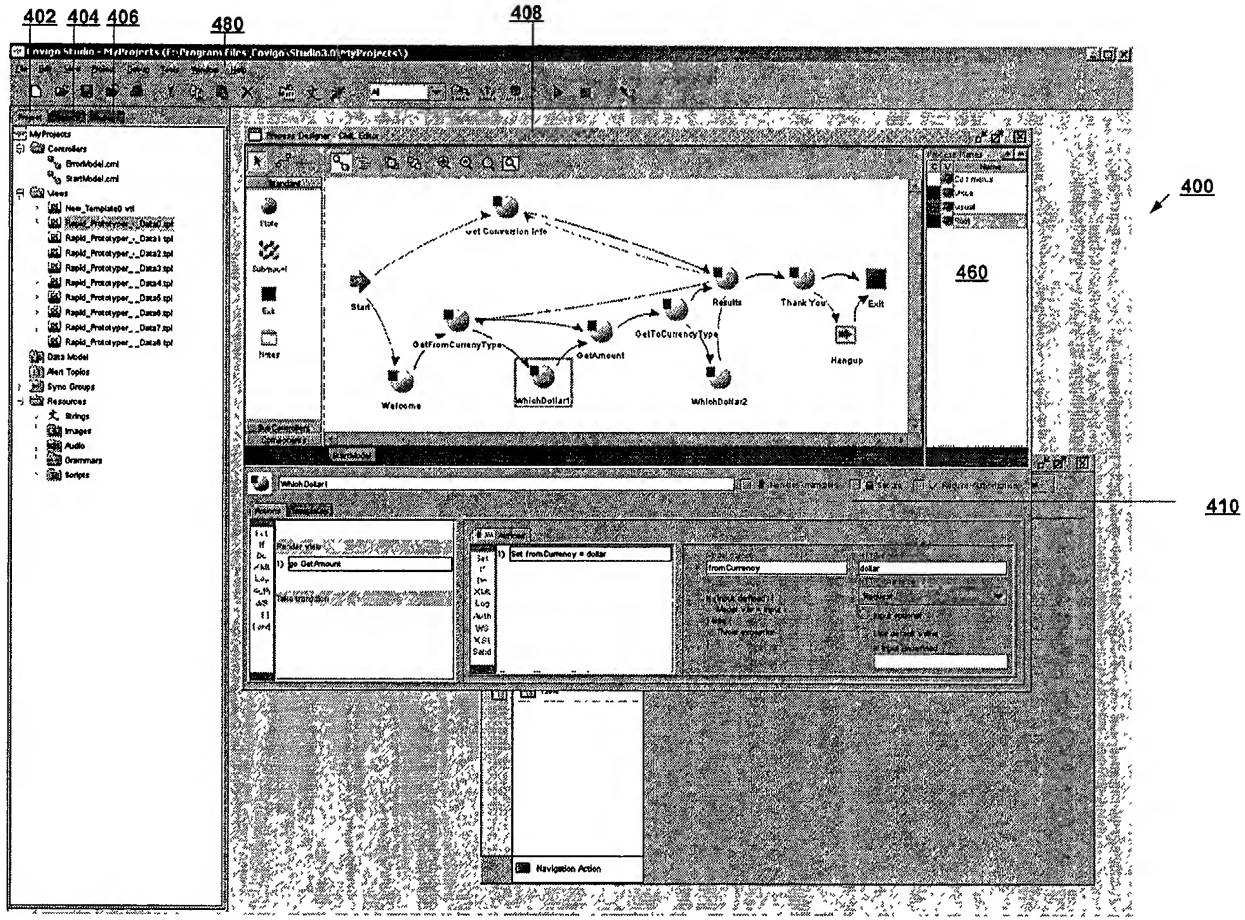
402
404
406
480
408
400
460
410
Start
Welcome
GetFromCurrencyType
GetAmount
WhichDollar1
GetToCurrencyType
WhichDollar2
Results
Thank You
Exit
Hangup
Navigation Action

Figure 17

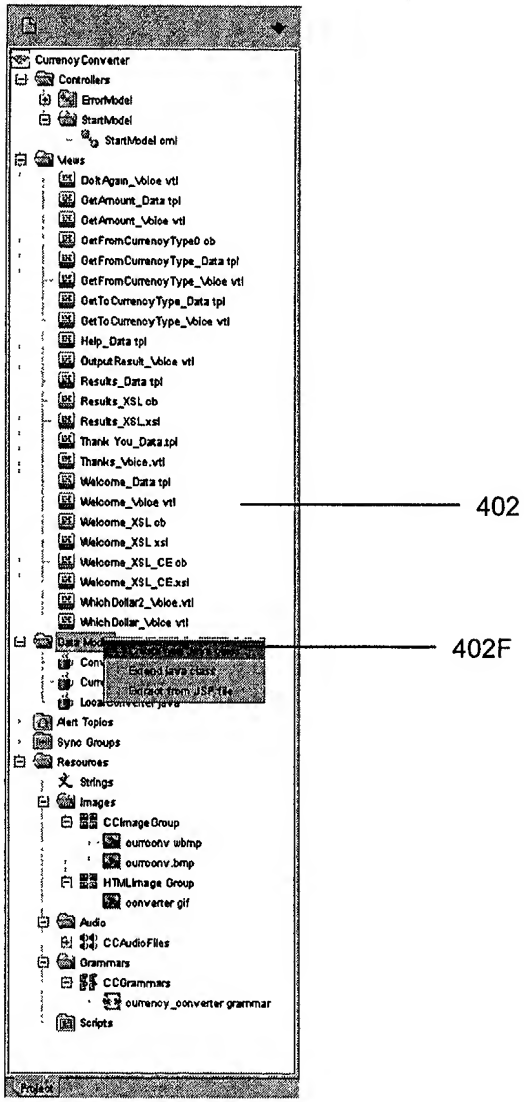
Currency Converter
Controllers
ErrorModel
StartModel
StartModel.eml
Views
DoItAgain_Voice.vtl
GetAmount_Data.tpl
GetAmount_Voice.vtl
GetFromCurrencyType0.ob
GetFromCurrencyType_Data.tpl
GetFromCurrencyType_Voice.vtl
GetToCurrencyType_Data.tpl
GetToCurrencyType_Voice.vtl
Help_Data.tpl
OutputResult_Voice.vtl
Results_Data.tpl
Results_XSL.ob
Results_XSL.xsl
Thank You_Data.tpl
Thanks_Voice.vtl
Welcome_Data.tpl
Welcome_Voice.vtl
Welcome_XSL.ob
Welcome_XSL.xsl
Welcome_XSL_CE.ob
Welcome_XSL_CE.xsl
WhichDollar2_Voice.vtl
WhichDollar_Voice.vtl
Alert Topics
Sync Groups
Resources
Strings
Images
CCImageGroup
ourroonv.wbmp
ourroonv.bmp
HTMLImage Group
converter.gif
Audio
CCAudioFiles
Grammars
CCGrammars
currency_converter.grammar
Scripts
402
402F

Figure 18

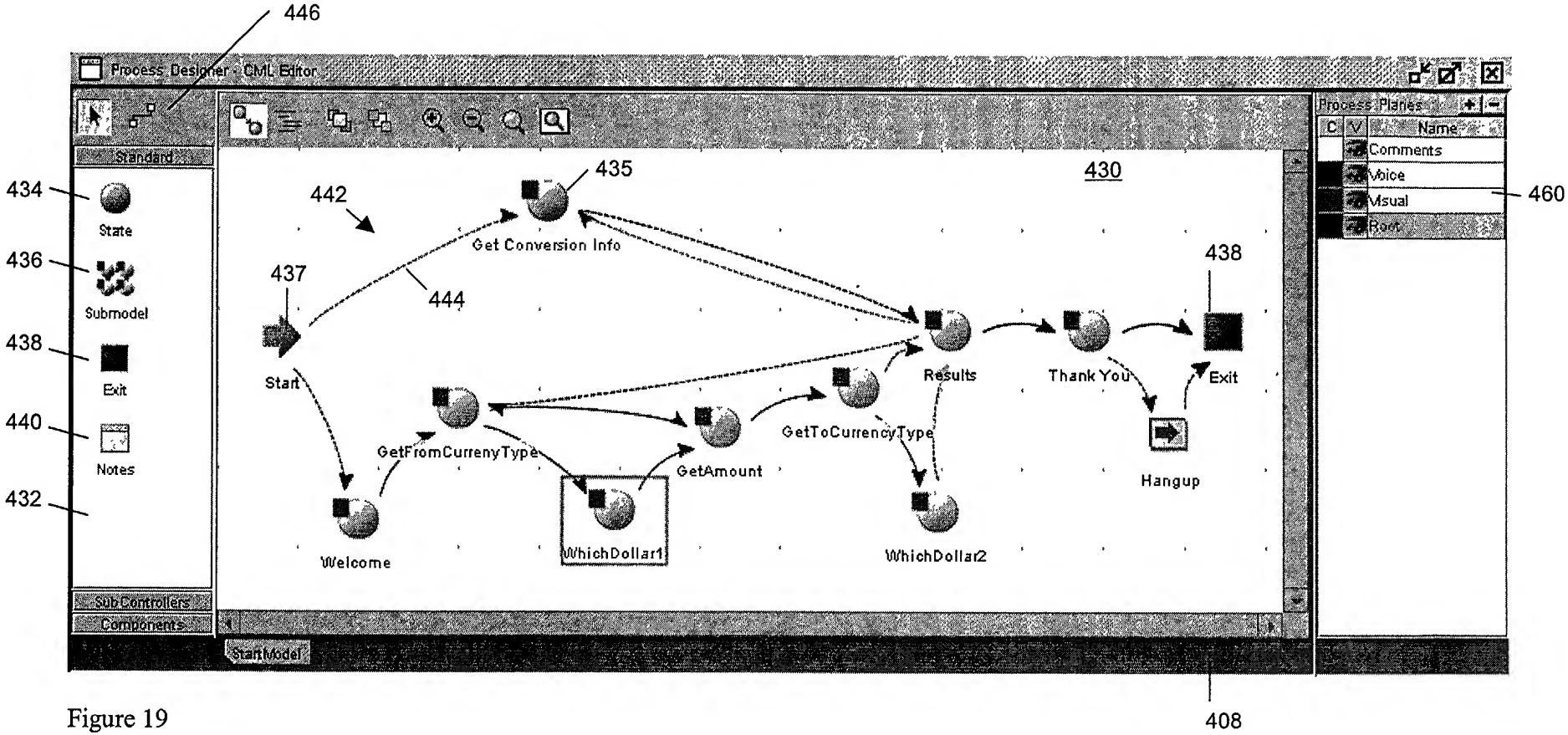
446
Process Designer - CML Editor
Standard
State
Submodel
Exit
Notes
SubControllers
Components
434
436
438
440
432
442
435
430
437
444
Get Conversion Info
Start
Results
Thank You
438
Exit
GetFromCurrenyType
GetAmount
GetToCurrencyType
Hangup
Welcome
WhichDollar1
WhichDollar2
StartModel
Process Planes
C
V
Name
Comments
Voice
Visual
Root
460
408

Figure 19

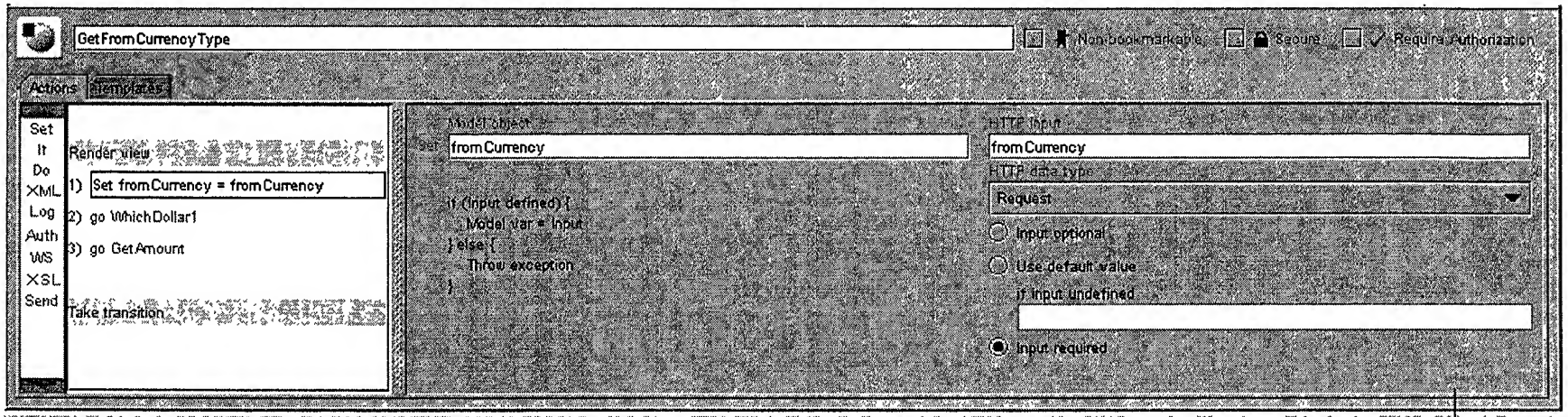
GetFromCurrencyType
Non-bookmarkable
Secure
Require Authorization
Actions
Templates
Set
It
Do
XML
Log
Auth
WS
XSL
Send
Render view
1) Set fromCurrency = fromCurrency
2) go WhichDollar1
3) go GetAmount
Take transition
Model object
Set
fromCurrency
if (input defined) {
Model.var = input
} else {
Throw exception
}
HTTP input
fromCurrency
HTTP data type
Request
Input optional
Use default value
if input undefined
Input required
410

Figure 20

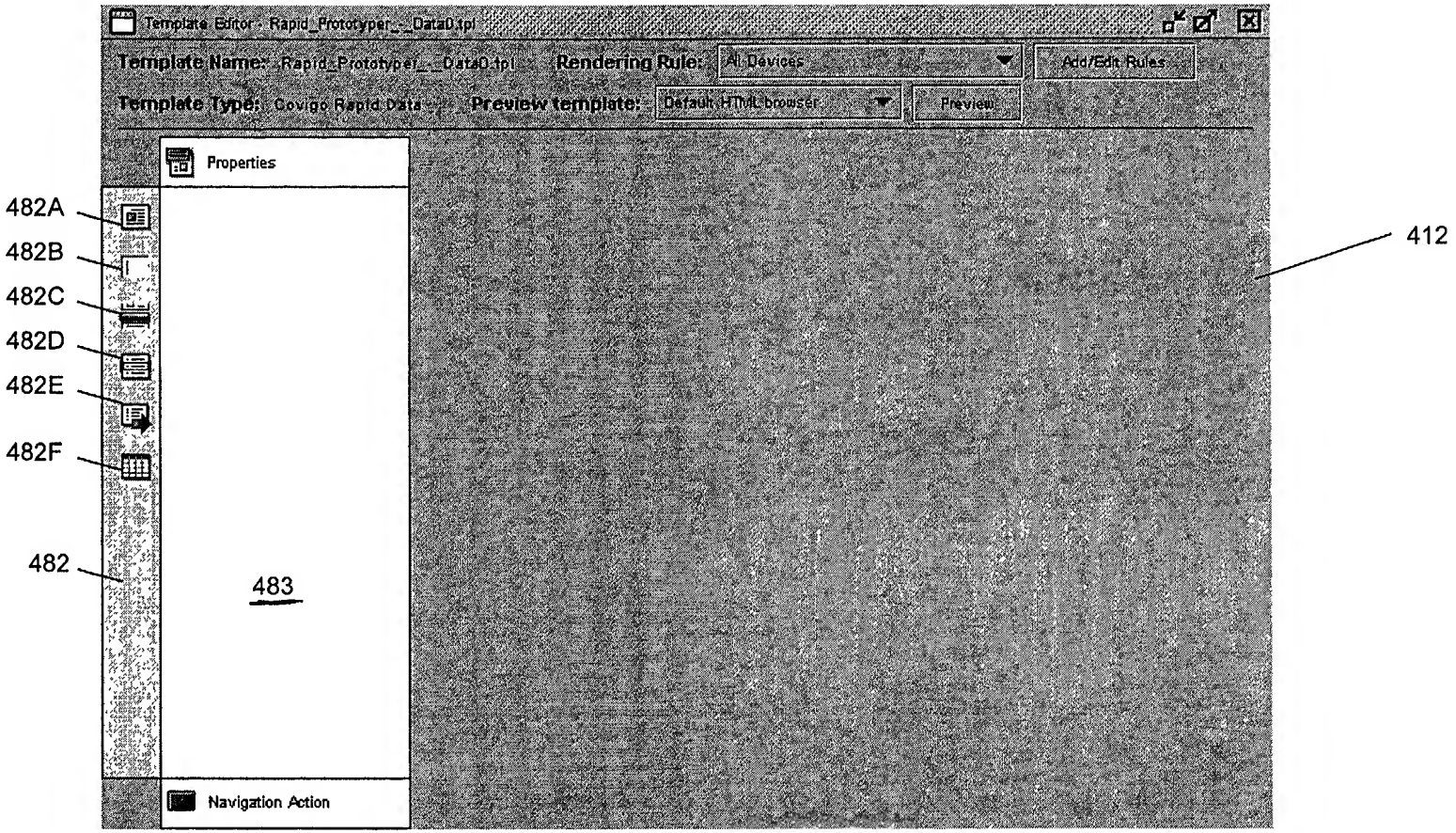
Template Editor - Rapid_Prototyper_-_Data0.tpl
Template Name: Rapid_Prototyper_-_Data0.tpl
Rendering Rule: All Devices
Add/Edit Rules
Template Type: Covigo Rapid Data
Preview template: Default HTML browser
Preview
Properties
Navigation Action
482A
482B
482C
482D
482E
482F
482
483
412

Figure 21

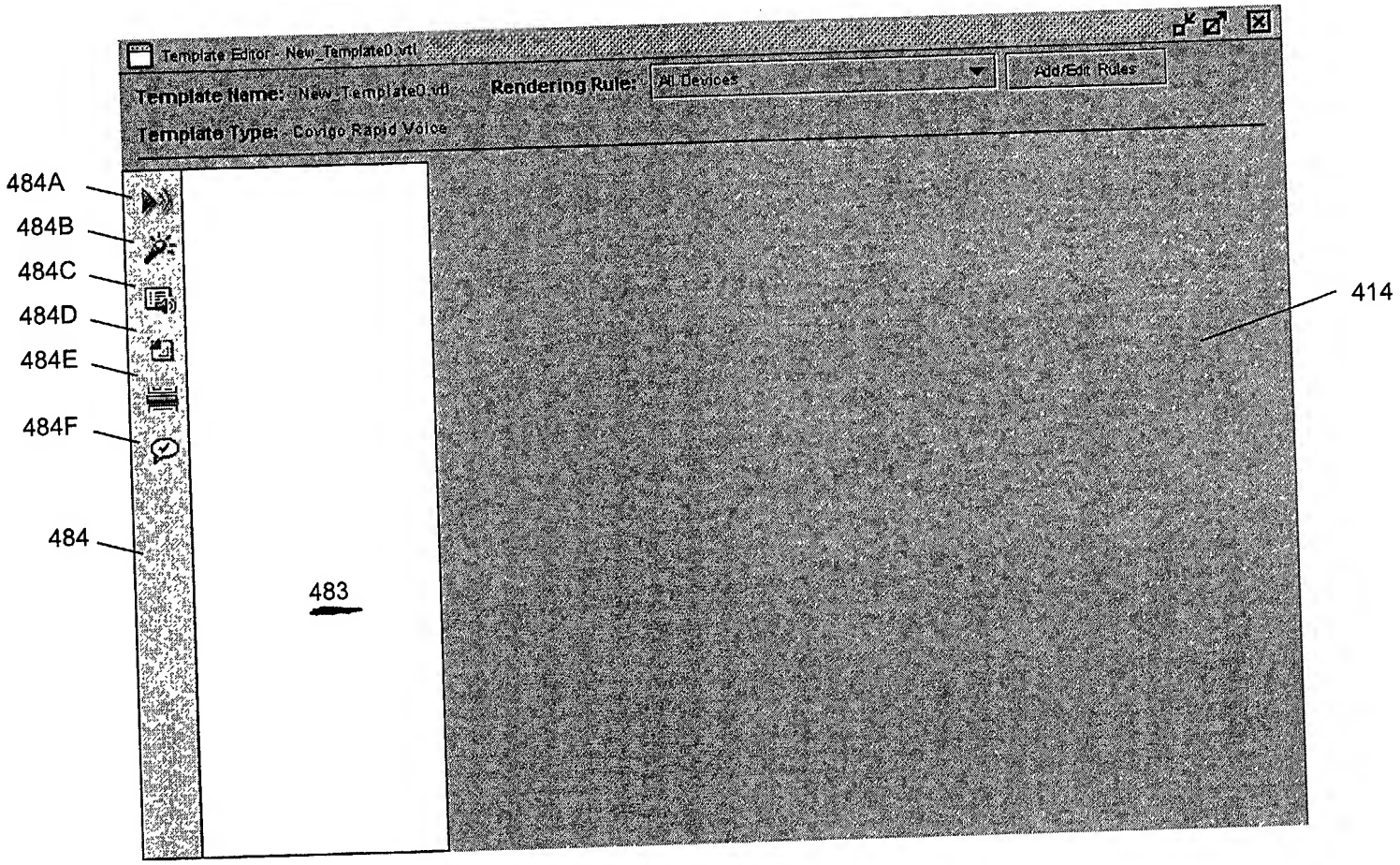
Template Editor - New_Template0.vtl
Template Name: New_Template0.vtl
Rendering Rule: All Devices
Add/Edit Rules
Template Type: Cordo Rapid Voice
484A
484B
484C
484D
484E
484F
484
483
414

Figure 22

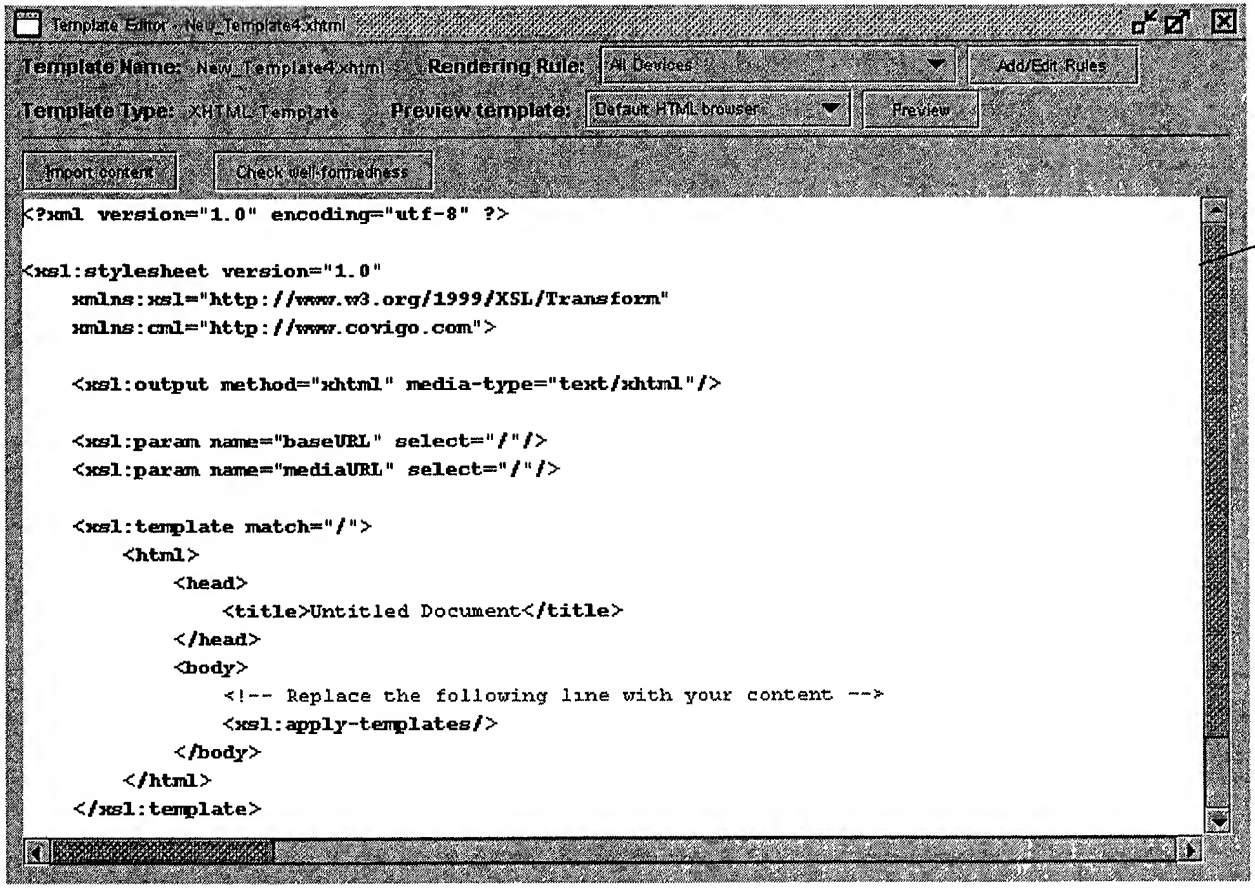
Template Editor - New_Template4.xhtml
Template Name: New_Template4.xhtml
Rendering Rule: All Devices
Add/Edit Rules
Template Type: XHTML Template
Preview template: Default HTML browser
Preview
Import content
Check well-formedness
<?xml version="1.0" encoding="utf-8" ?>
<xsl:stylesheet version="1.0"
xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
xmlns:cml="http://www.covigo.com">
<xsl:output method="xhtml" media-type="text/xhtml"/>
<xsl:param name="baseURL" select="/"/>
<xsl:param name="mediaURL" select="/"/>
<xsl:template match="/">
<html>
<head>
<title>Untitled Document</title>
</head>
<body>
<!-- Replace the following line with your content -->
<xsl:apply-templates/>
</body>
</html>
</xsl:template>
416

Figure 23

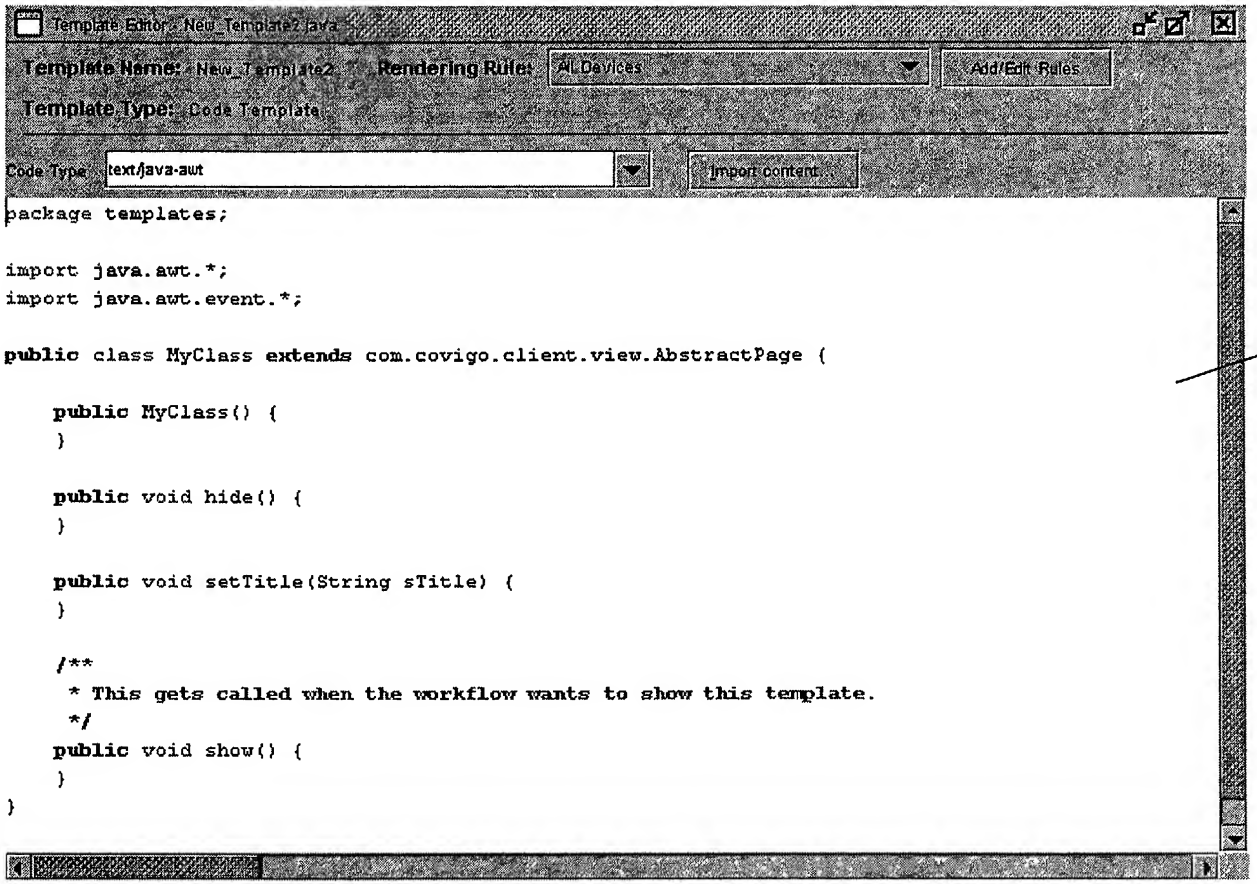
Template Editor - New_Template2.java
Template Name: New_Template2
Rendering Rule: All Devices
Add/Edit Rules
Template Type: Code Template
Code Type: text/java-aut
Import content...
package templates;
import java.awt.*;
import java.awt.event.*;
public class MyClass extends com.covigo.client.view.AbstractPage {
public MyClass() {
}
public void hide() {
}
public void setTitle(String sTitle) {
}
/**
* This gets called when the workflow wants to show this template.
*/
public void show() {
}
}
418

Figure 24

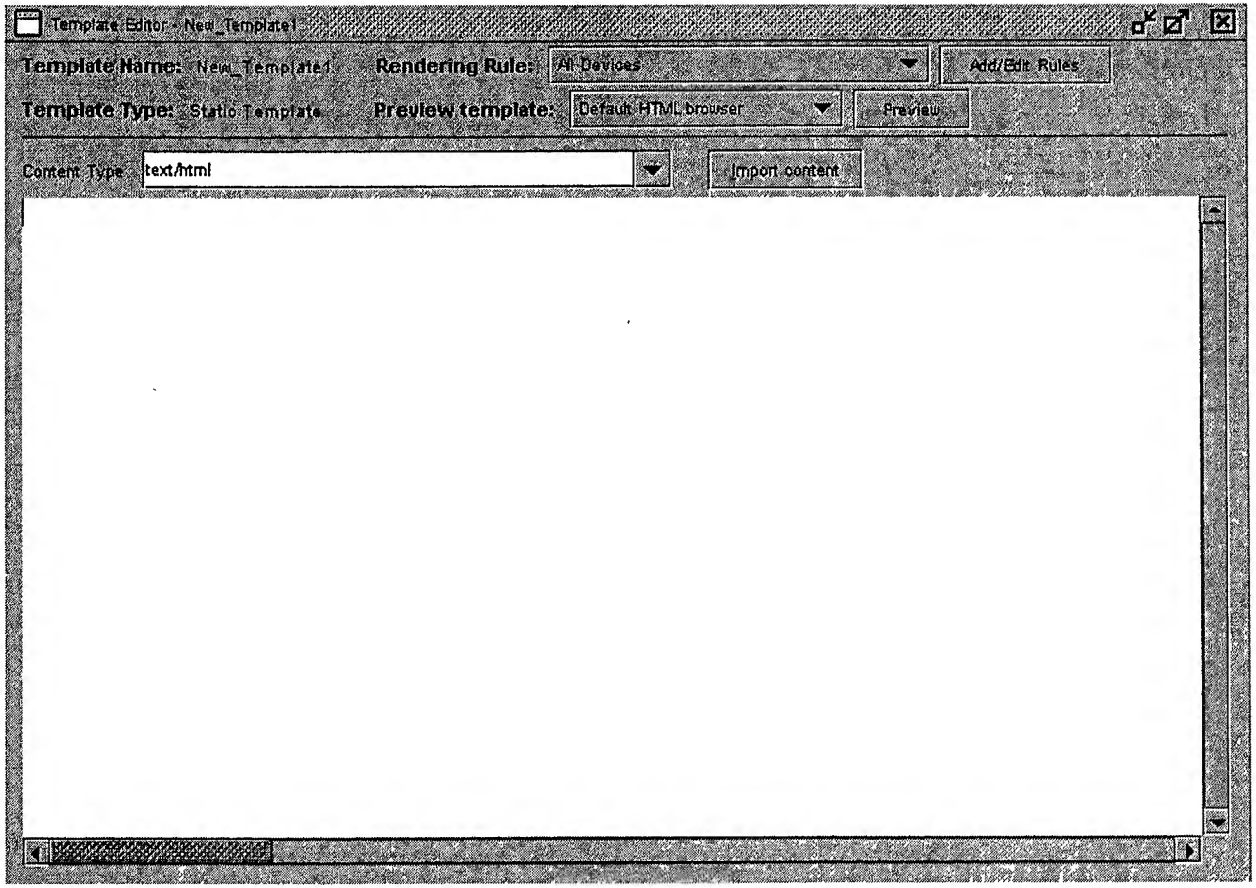
Template Editor - New_Template1
Template Name: New_Template1
Rendering Rule: All Devices
Add/Edit Rules
Template Type: Static Template
Preview template: Default HTML browser
Preview
Content Type: text/html
Import content
420

Figure 25

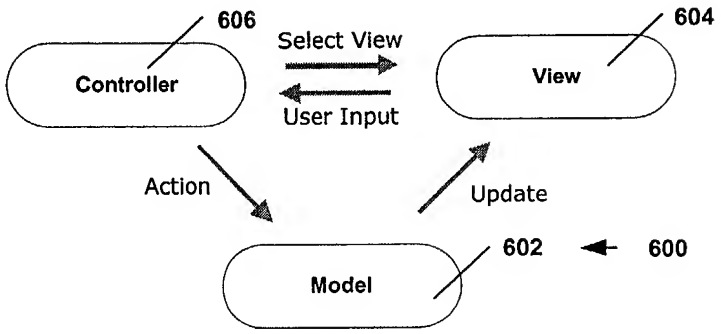

Figure 26 (Prior Art)

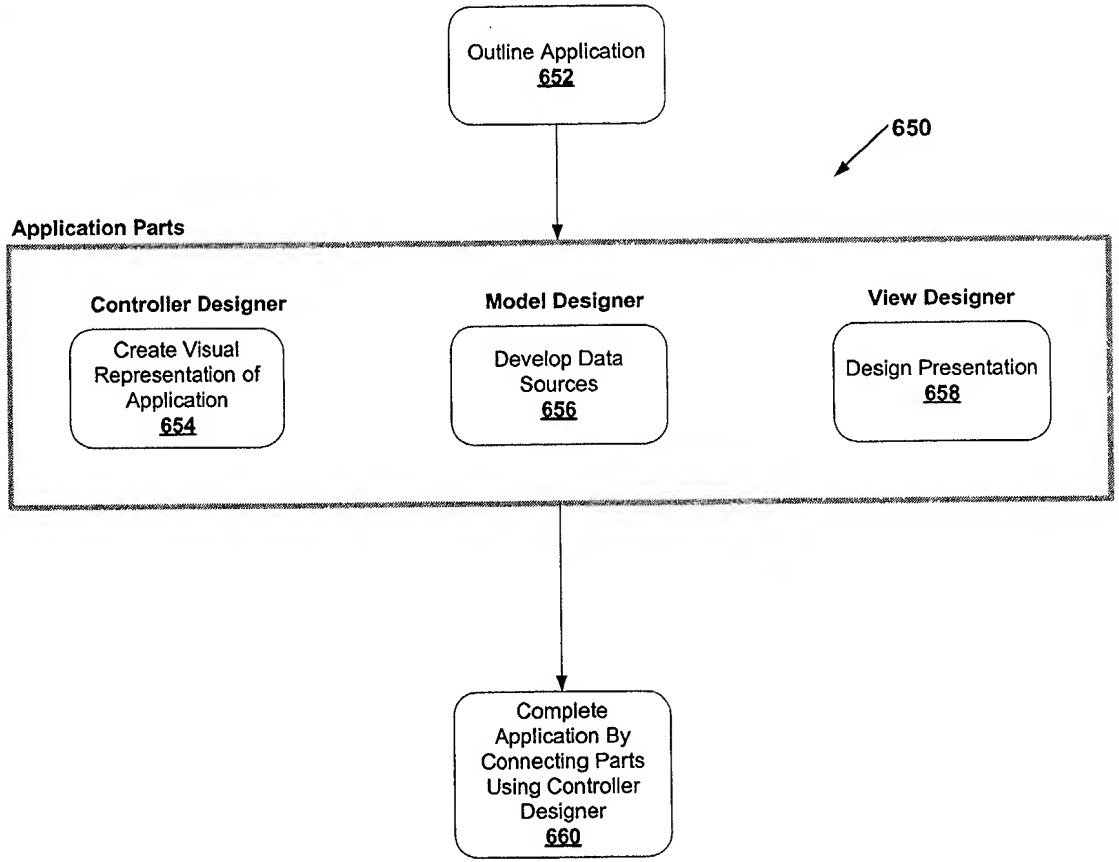
Outline Application
652
650
Application Parts
Controller Designer
Create Visual Representation of Application
654
Model Designer
Develop Data Sources
656
View Designer
Design Presentation
658
Complete Application By Connecting Parts Using Controller Designer
660

Figure 27

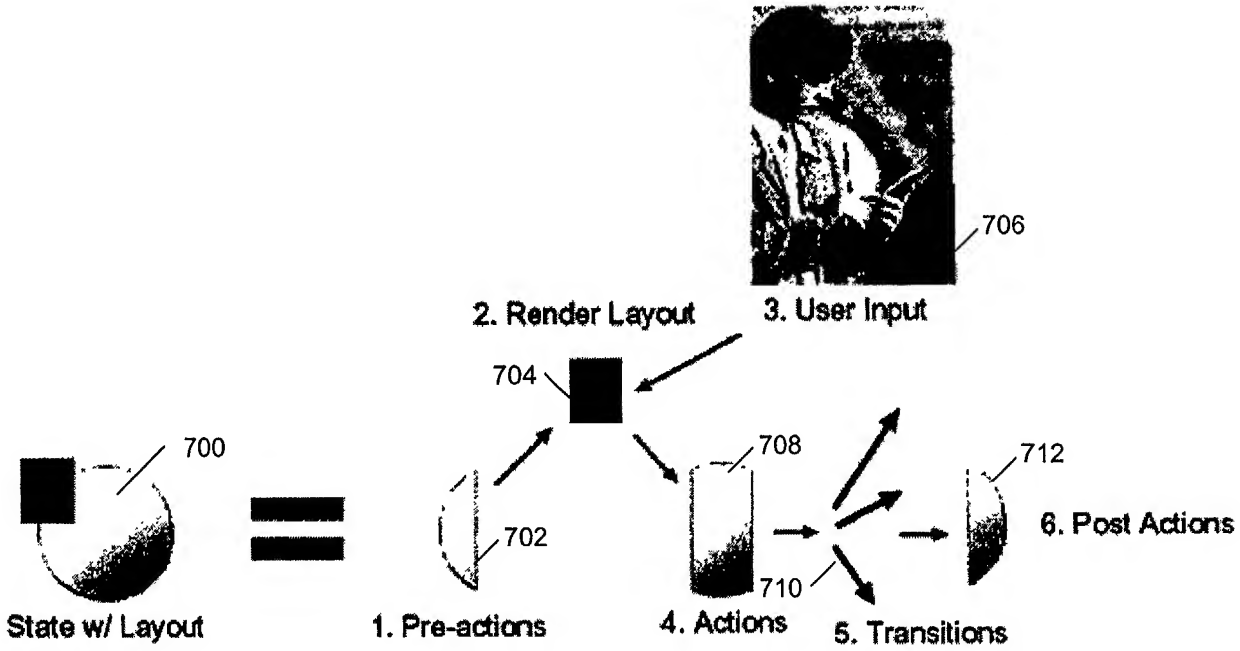
706
2. Render Layout
3. User Input
704
700
708
712
6. Post Actions
702
710
State w/ Layout
1. Pre-actions
4. Actions
5. Transitions

Figure 28

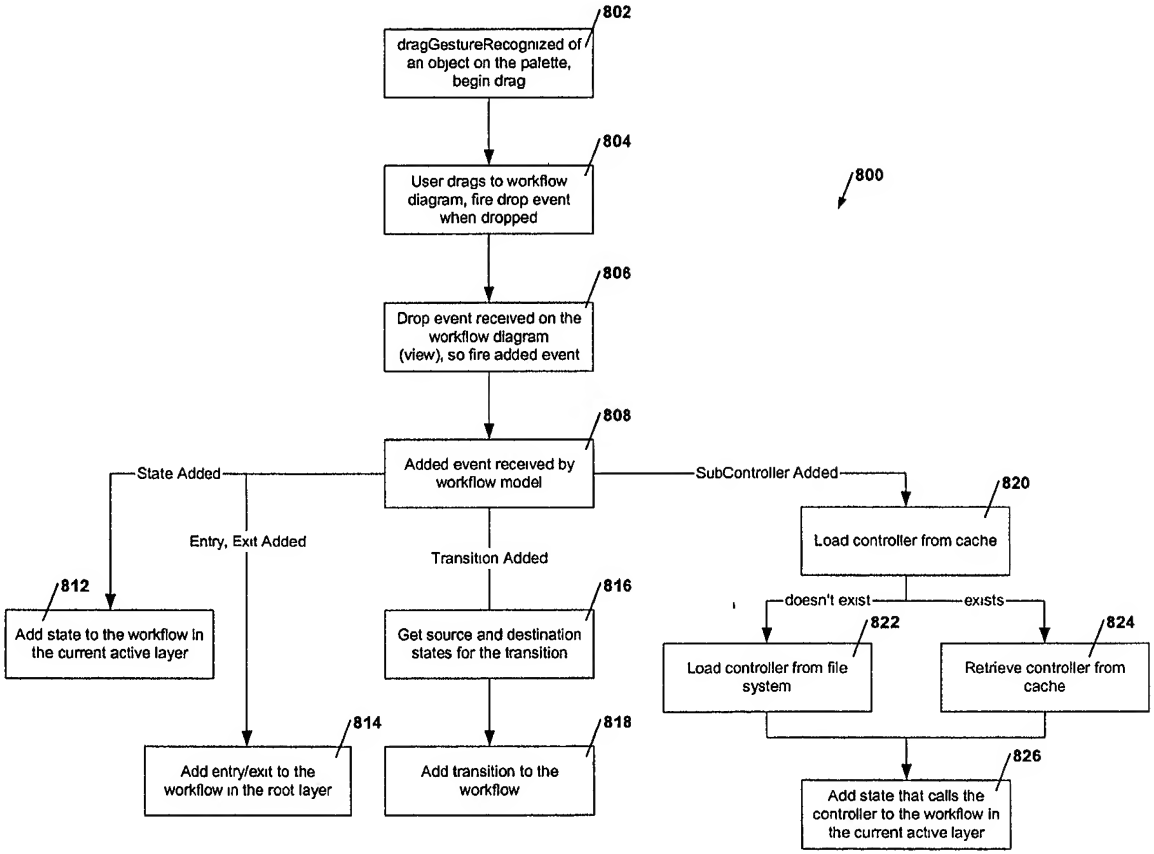
802
dragGestureRecognized of an object on the palette, begin drag
804
User drags to workflow diagram, fire drop event when dropped
800
806
Drop event received on the workflow diagram (view), so fire added event
808
Added event received by workflow model
State Added
Entry, Exit Added
Transition Added
SubController Added
812
Add state to the workflow in the current active layer
814
Add entry/exit to the workflow in the root layer
816
Get source and destination states for the transition
818
Add transition to the workflow
820
Load controller from cache
doesn't exist
exists
822
Load controller from file system
824
Retrieve controller from cache
826
Add state that calls the controller to the workflow in the current active layer

Figure 29

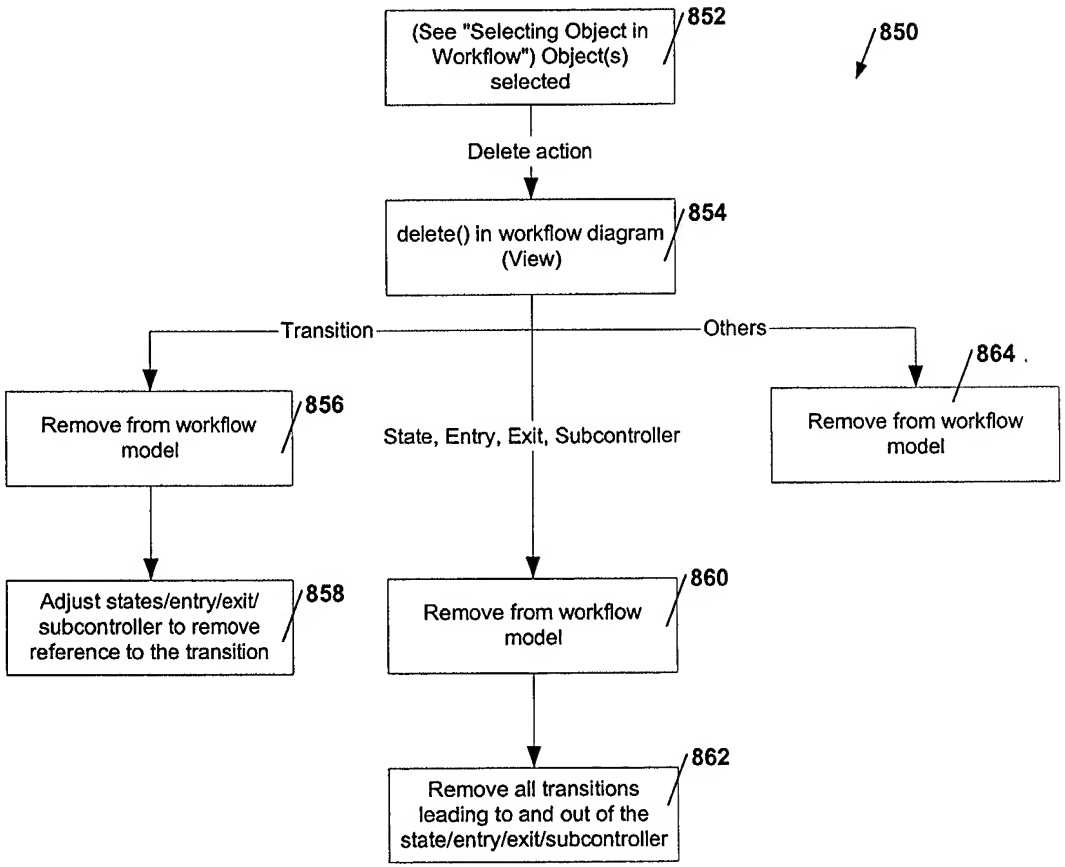
(See "Selecting Object in Workflow") Object(s) selected
852
850
Delete action
delete() in workflow diagram (View)
854
Transition
Others
State, Entry, Exit, Subcontroller
Remove from workflow model
856
Remove from workflow model
864
Adjust states/entry/exit/ subcontroller to remove reference to the transition
858
Remove from workflow model
860
Remove all transitions leading to and out of the state/entry/exit/subcontroller
862

Figure 30

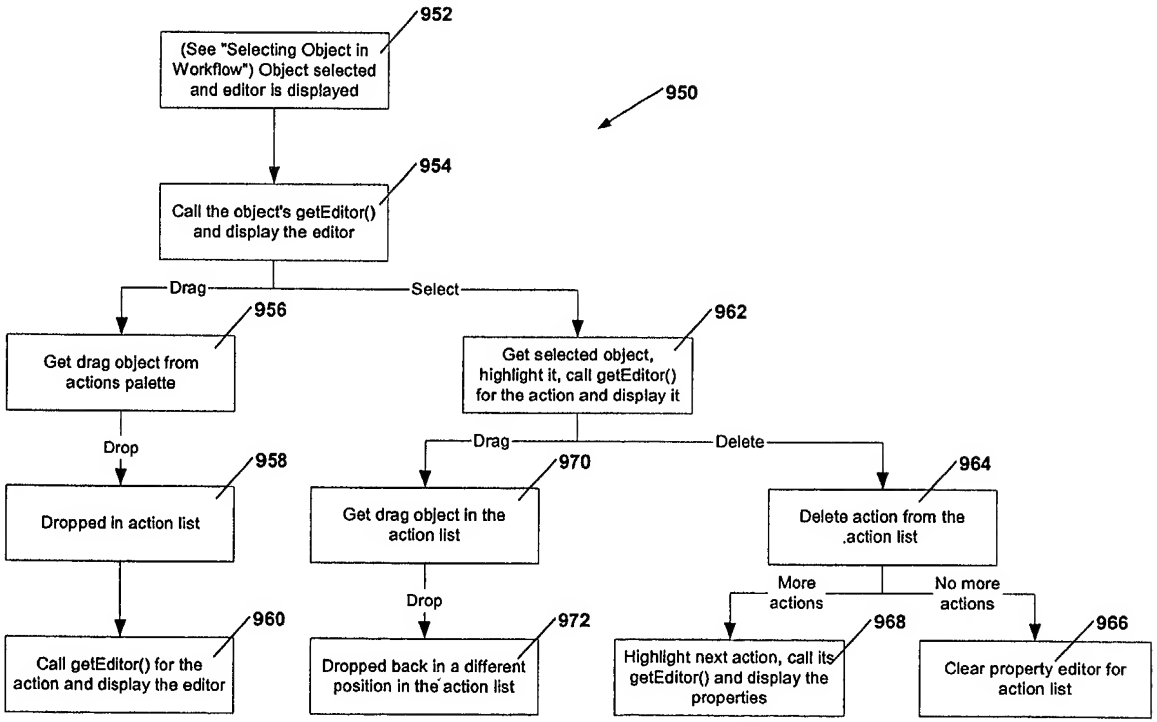
952
(See "Selecting Object in Workflow") Object selected and editor is displayed
950
954
Call the object's getEditor() and display the editor
Drag
Select
956
Get drag object from actions palette
962
Get selected object, highlight it, call getEditor() for the action and display it
Drop
Drag
Delete
958
Dropped in action list
970
Get drag object in the action list
964
Delete action from the action list
Drop
More actions
No more actions
960
Call getEditor() for the action and display the editor
972
Dropped back in a different position in the action list
968
Highlight next action, call its getEditor() and display the properties
966
Clear property editor for action list

Figure 31

Patent Application

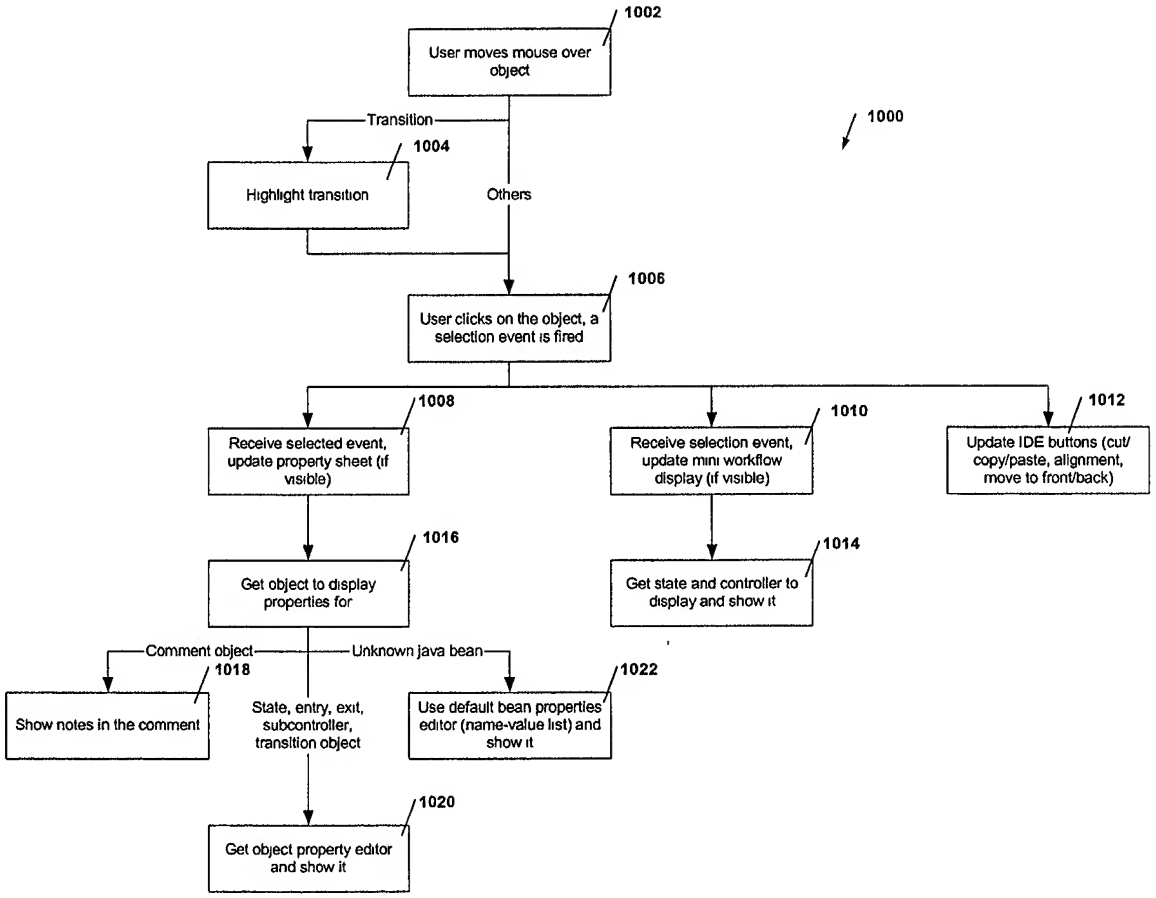

Figure 32

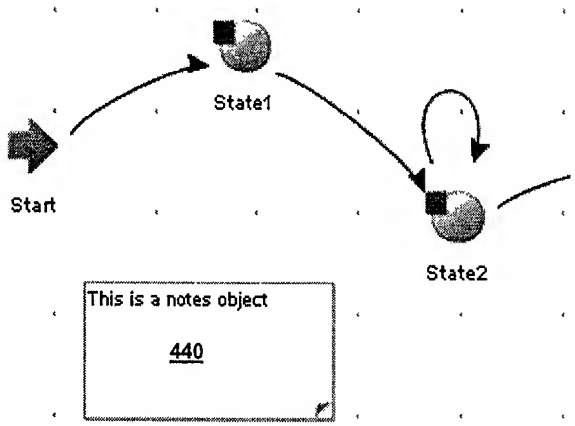

Figure 33

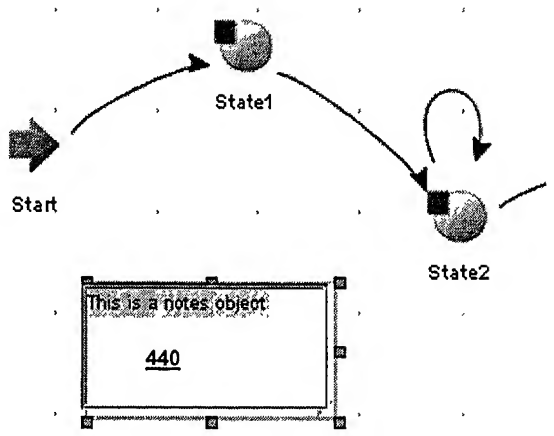

Figure 34

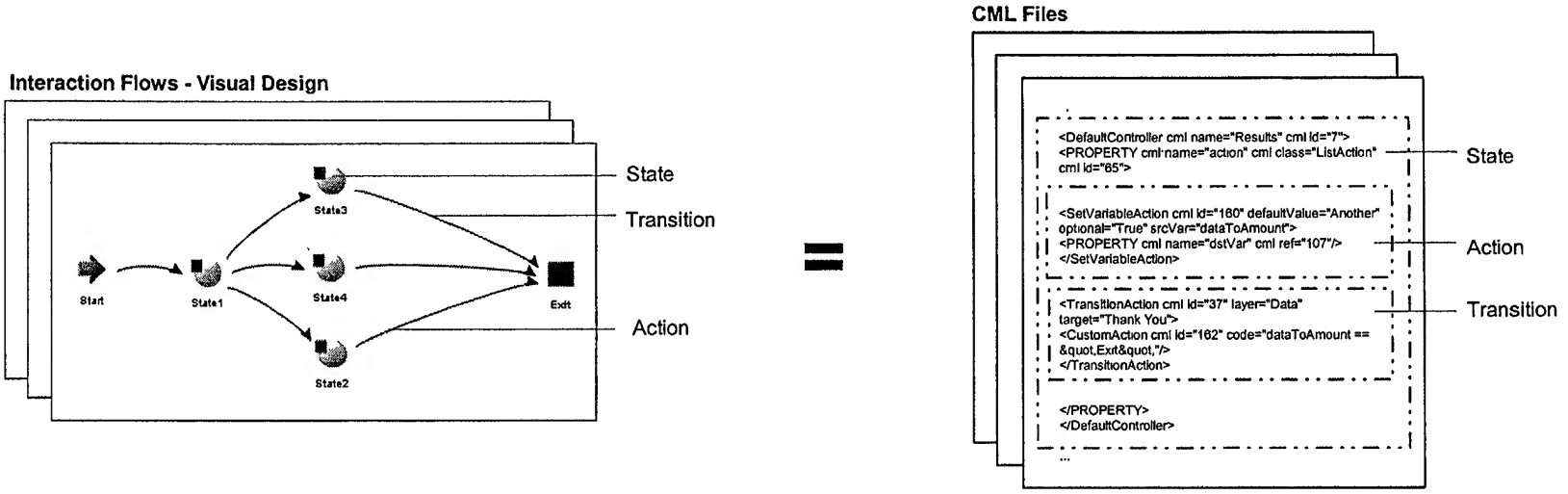
Interaction Flows - Visual Design
State
Transition
Action
Start
State1
State3
State4
State2
Exit
=
CML Files
<DefaultController cml name="Results" cml id="7">
<PROPERTY cml name="action" cml class="ListAction"
cml id="65">
<SetVariableAction cml id="160" defaultValue="Another"
optional="True" srcVar="dataToAmount">
<PROPERTY cml name="dstVar" cml ref="107"/>
</SetVariableAction>
<TransitionAction cml id="37" layer="Data"
target="Thank You">
<CustomAction cml id="162" code="dataToAmount ==
&quot,Exit&quot,"/>
</TransitionAction>
</PROPERTY>
</DefaultController>
...
State
Action
Transition

Figure 35

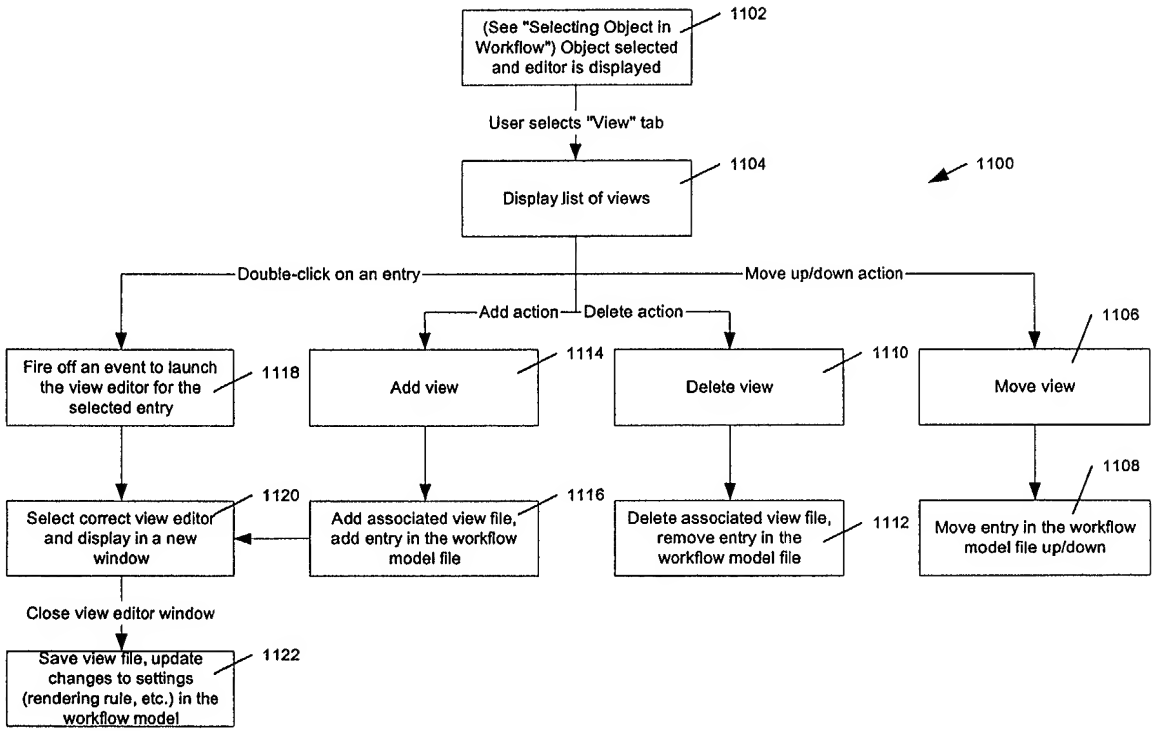
(See "Selecting Object in Workflow") Object selected and editor is displayed
1102
User selects "View" tab
Display list of views
1104
1100
Double-click on an entry
Add action
Delete action
Move up/down action
1106
Fire off an event to launch the view editor for the selected entry
1118
Add view
1114
Delete view
1110
Move view
1108
Select correct view editor and display in a new window
1120
Add associated view file, add entry in the workflow model file
1116
Delete associated view file, remove entry in the workflow model file
1112
Move entry in the workflow model file up/down
Close view editor window
Save view file, update changes to settings (rendering rule, etc.) in the workflow model
1122

Figure 36

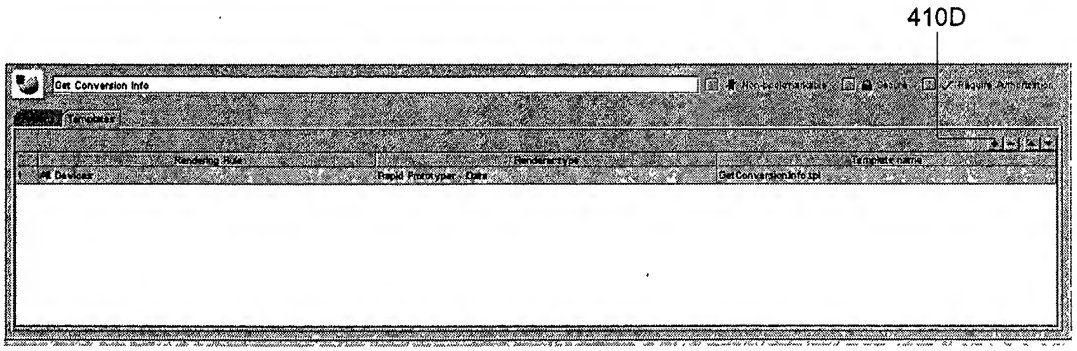

Figure 37

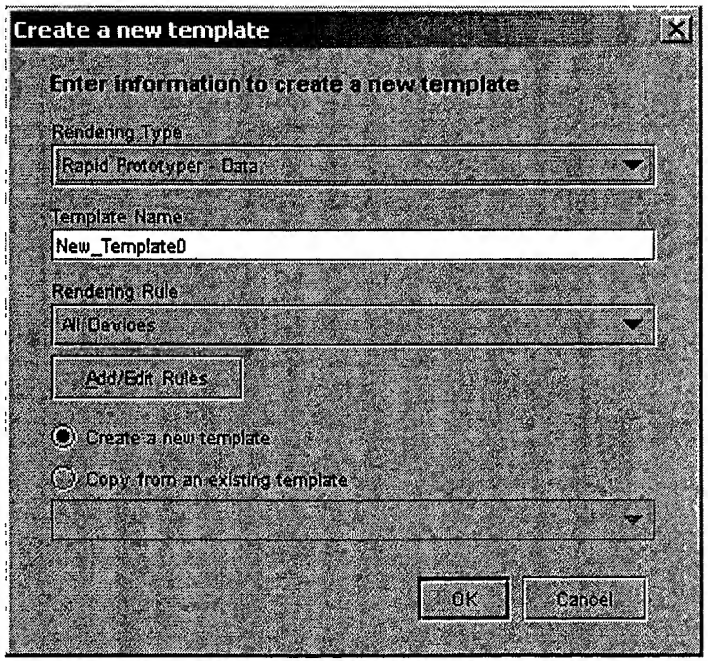

Figure 38

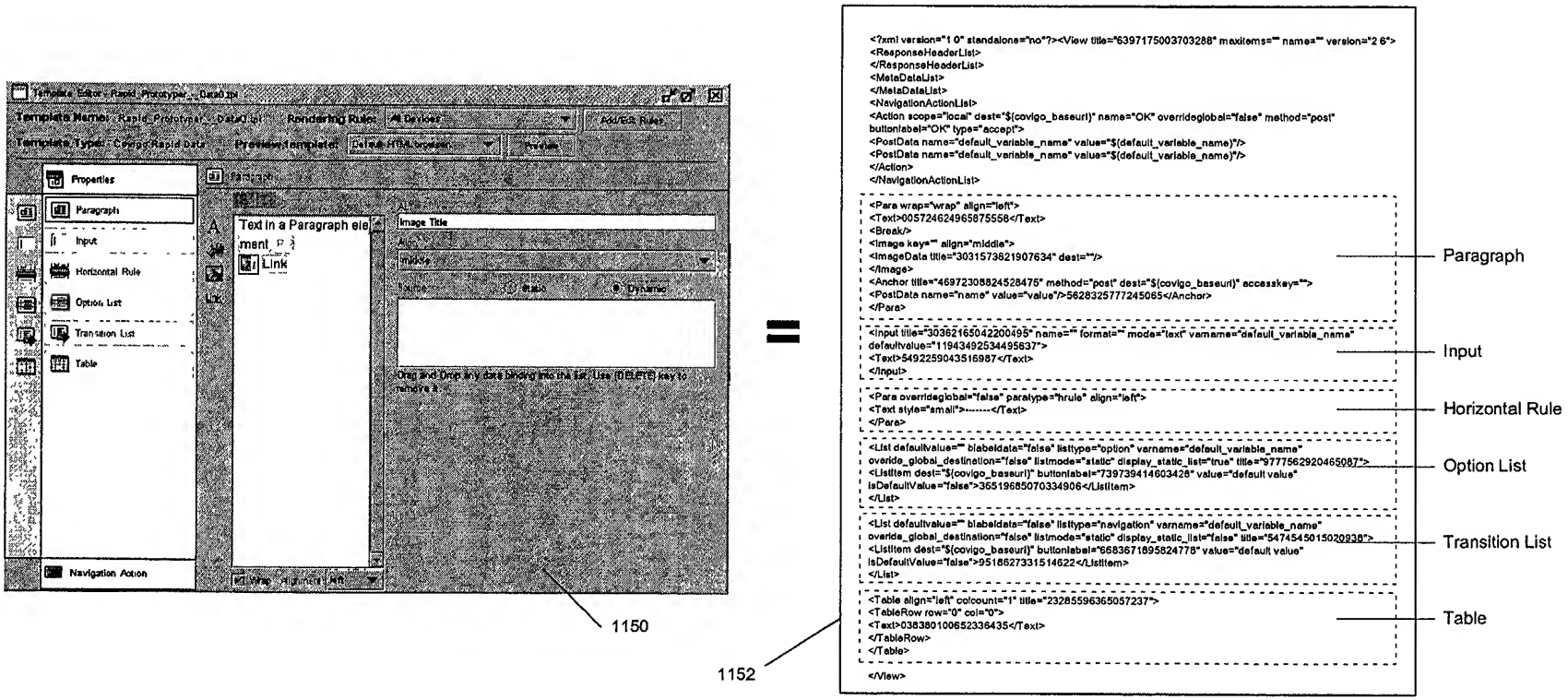
Properties
Paragraph
Input
Horizontal Rule
Option List
Transition List
Table
Navigation Action
Text in a Paragraph ele
ment
Link
Image Title
1150
1152
<?xml version="1 0" standalone="no"?><View title="6397175003703288" maxitems="" name="" version="2 6">
<ResponseHeaderList>
</ResponseHeaderList>
<MetaDataList>
</MetaDataList>
<NavigationActionList>
<Action scope="local" dest="${covigo_baseurl}" name="OK" overrideglobal="false" method="post"
buttonlabel="OK" type="accept">
<PostData name="default_variable_name" value="${default_variable_name}"/>
<PostData name="default_variable_name" value="${default_variable_name}"/>
</Action>
</NavigationActionList>
<Para wrap="wrap" align="left">
<Text>0057246249658755 58</Text>
<Break/>
<Image key="" align="middle">
<ImageData title="3031573821907634" dest=""/>
</Image>
<Anchor title="46972308824528475" method="post" dest="${covigo_baseurl}" accesskey="">
<PostData name="name" value="value"/>5628325777245065</Anchor>
</Para>
<Input title="303621650422004 95" name="" format="" mode="text" varname="default_variable_name"
defaultvalue="11943492534495637">
<Text>5492259043516987</Text>
</Input>
<Para overrideglobal="false" paratype="hrule" align="left">
<Text style="small">-------</Text>
</Para>
<List defaultvalue="" blabeldata="false" listtype="option" varname="default_variable_name"
overide_global_destination="false" listmode="static" display_static_list="true" title="9777562920465087">
<ListItem dest="${covigo_baseurl}" buttonlabel="739739414603428" value="default value"
isDefaultValue="false">365196850703349 06</ListItem>
</List>
<List defaultvalue="" blabeldata="false" listtype="navigation" varname="default_variable_name"
overide_global_destination="false" listmode="static" display_static_list="false" title="5474545015020938">
<ListItem dest="${covigo_baseurl}" buttonlabel="668367189582477 8" value="default value"
isDefaultValue="false">9518627331514622</ListItem>
</List>
<Table align="left" colcount="1" title="23285596365057237">
<TableRow row="0" col="0">
<Text>0383801006523364 35</Text>
</TableRow>
</Table>
</View>
Paragraph
Input
Horizontal Rule
Option List
Transition List
Table

Figure 39

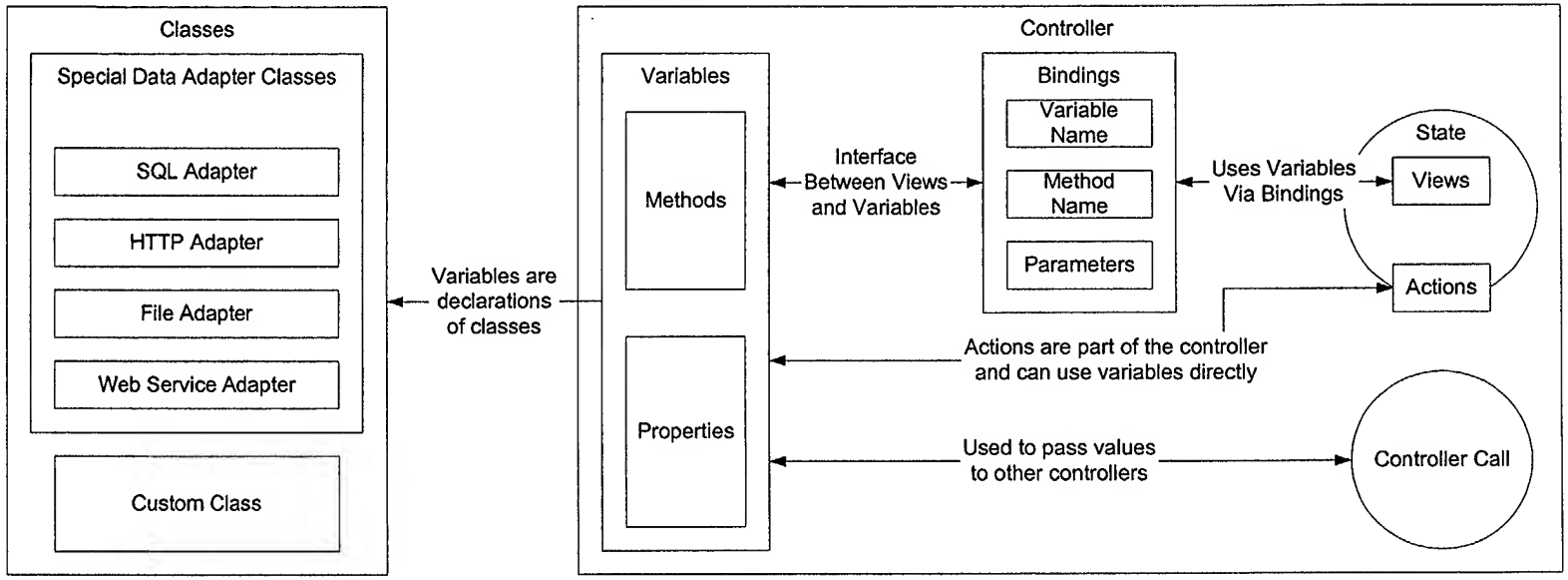
Classes
Special Data Adapter Classes
SQL Adapter
HTTP Adapter
File Adapter
Web Service Adapter
Custom Class
Variables are declarations of classes
Controller
Variables
Methods
Properties
Interface Between Views and Variables
Bindings
Variable Name
Method Name
Parameters
Uses Variables Via Bindings
State
Views
Actions
Actions are part of the controller and can use variables directly
Used to pass values to other controllers
Controller Call

Figure 40

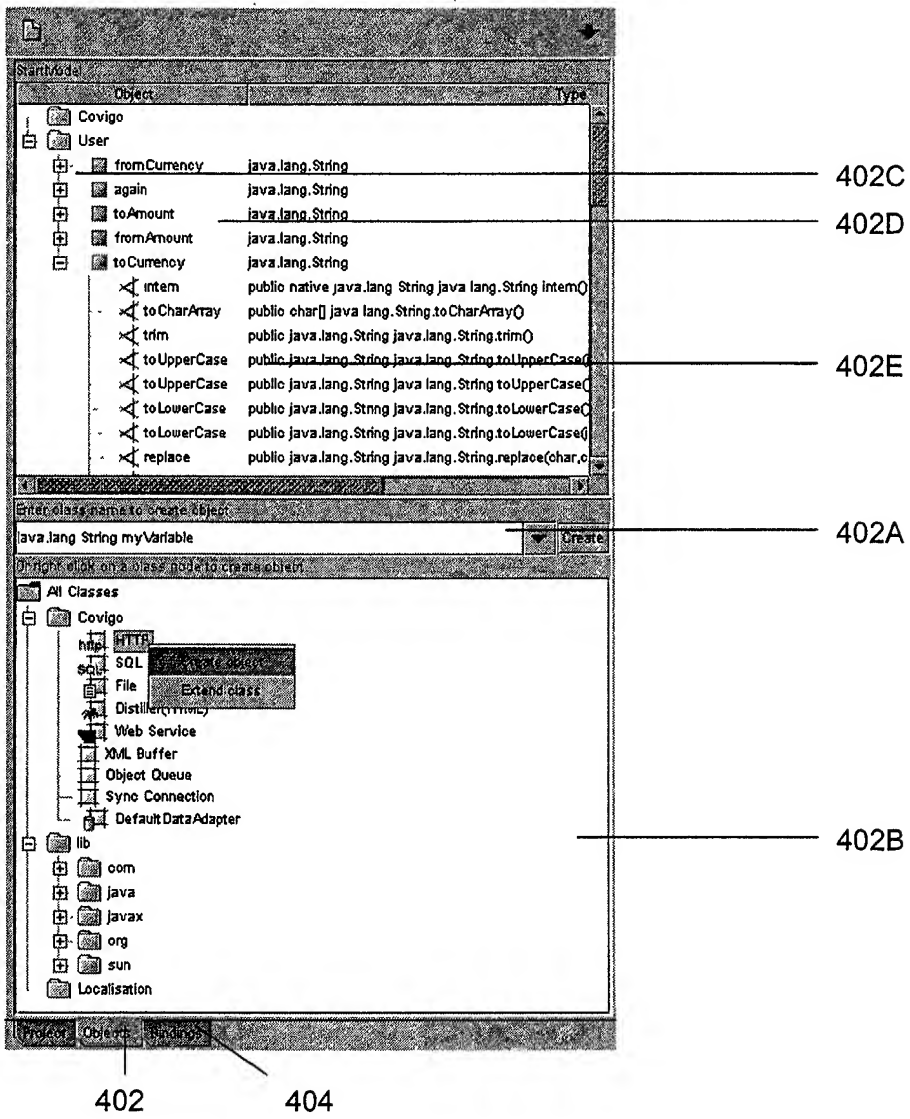
StartModel
Object
Type
Covigo
User
fromCurrency java.lang.String
again java.lang.String
toAmount java.lang.String
fromAmount java.lang.String
toCurrency java.lang.String
intern public native java.lang.String java.lang.String intern()
toCharArray public char[] java.lang.String.toCharArray()
trim public java.lang.String java.lang.String.trim()
toUpperCase public java.lang.String java.lang.String.toUpperCase()
toUpperCase public java.lang.String java.lang.String.toUpperCase()
toLowerCase public java.lang.String java.lang.String.toLowerCase()
toLowerCase public java.lang.String java.lang.String.toLowerCase()
replace public java.lang.String java.lang.String.replace(char,c
Enter class name to create object
java.lang.String myVariable
Create
Or right click on a class node to create object
All Classes
Covigo
HTTP
SQL
File
Extend class
Web Service
XML Buffer
Object Queue
Sync Connection
DefaultDataAdapter
lib
com
java
javax
org
sun
Localisation
Project
Objects
Bindings
402C
402D
402E
402A
402B
402
404

Figure 41

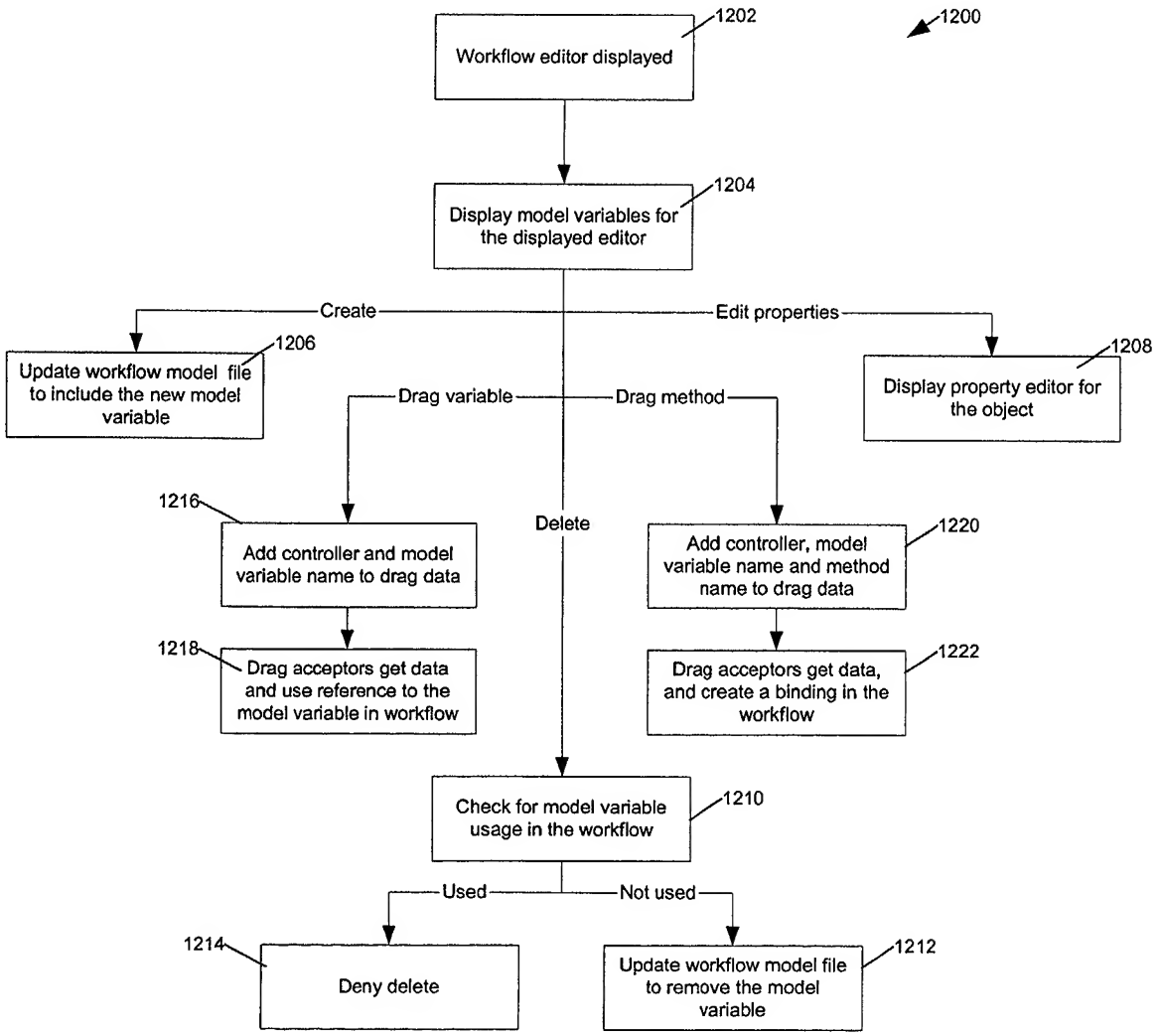
1200
1202
Workflow editor displayed
1204
Display model variables for the displayed editor
Create
Edit properties
1206
Update workflow model file to include the new model variable
1208
Display property editor for the object
Drag variable
Drag method
Delete
1216
Add controller and model variable name to drag data
1220
Add controller, model variable name and method name to drag data
1218
Drag acceptors get data and use reference to the model variable in workflow
1222
Drag acceptors get data, and create a binding in the workflow
1210
Check for model variable usage in the workflow
Used
Not used
1214
Deny delete
1212
Update workflow model file to remove the model variable

Figure 42

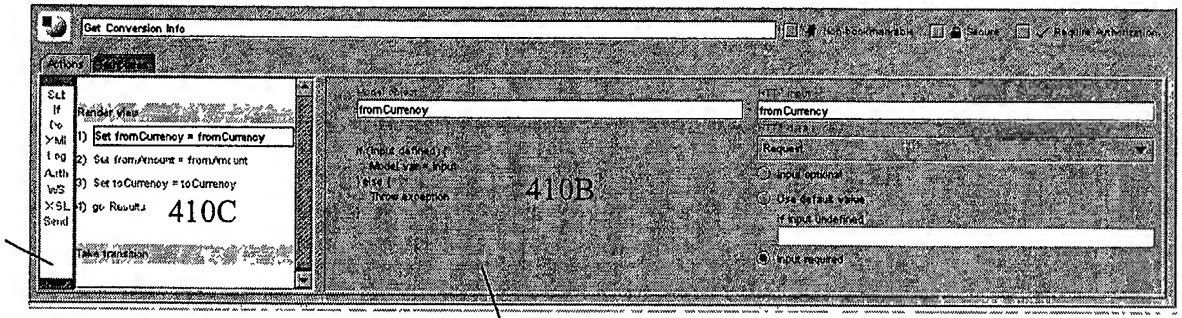

Figure 43

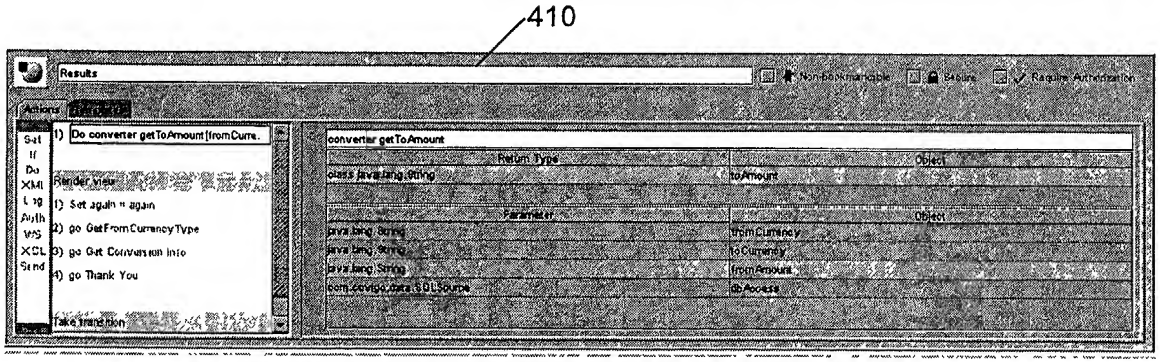

Figure 44

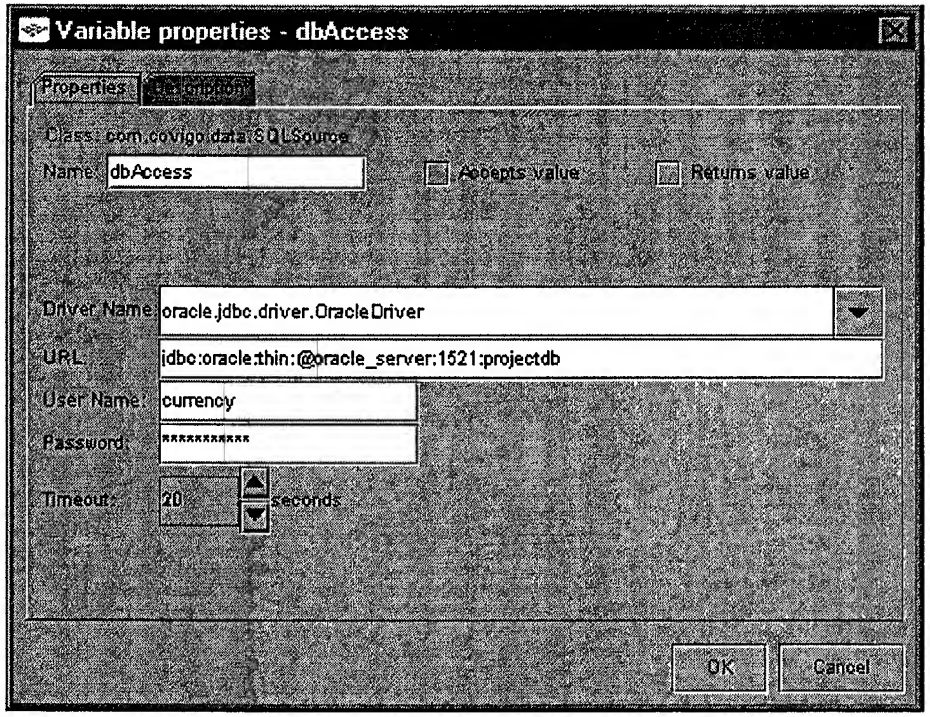
Variable properties - dbAccess
Properties
Description
Class: com.covigo.data.SQLSource
Name: dbAccess
Accepts value
Returns value
Driver Name: oracle.jdbc.driver.OracleDriver
URL: jdbc:oracle:thin:@oracle_server:1521:projectdb
User Name: currency
Password: **********
Timeout: 20 seconds
OK
Cancel

Figure 45

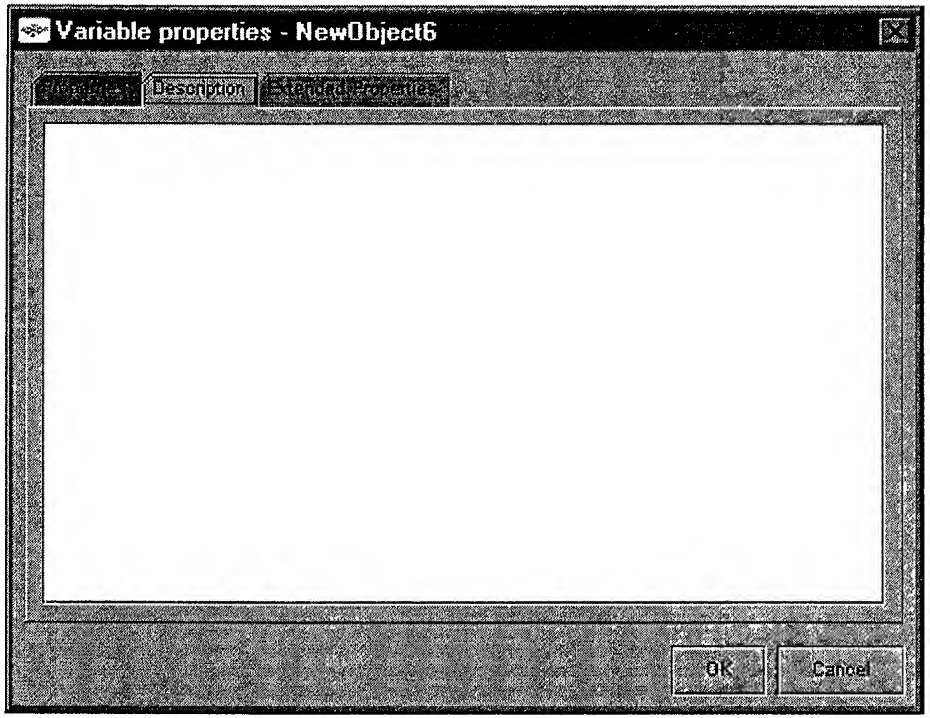

Figure 46

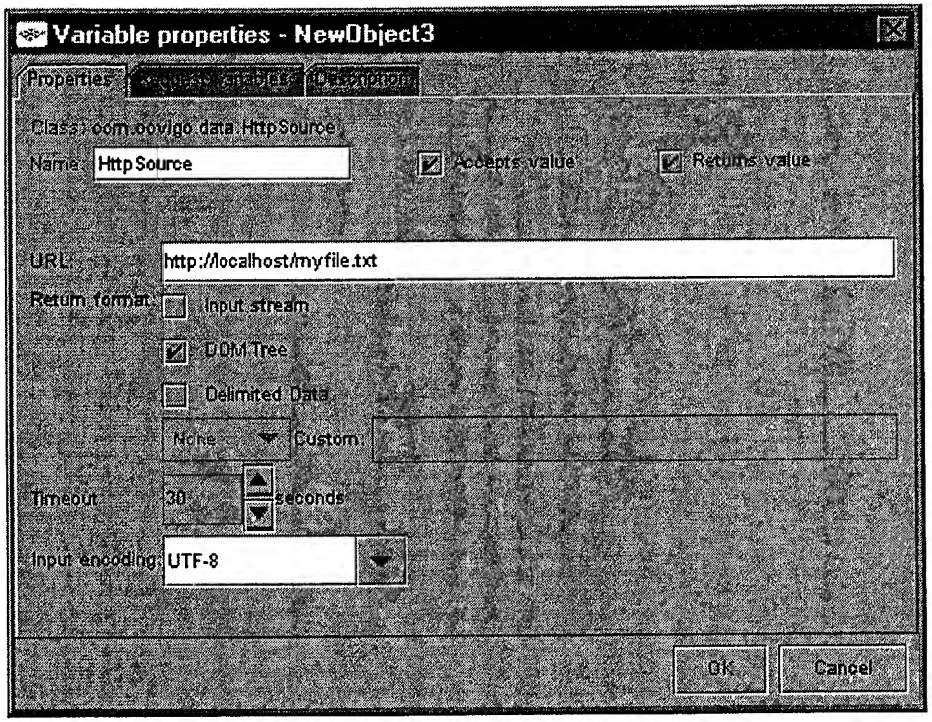
Variable properties - NewObject3
Properties
Request variables
Description
Class: com.oovigo.data.HttpSource
Name: HttpSource
Accepts value
Returns value
URL:
http://localhost/myfile.txt
Return format:
Input stream
DOM Tree
Delimited Data
None
Custom:
Timeout:
30
seconds
Input encoding:
UTF-8
OK
Cancel

Figure 47

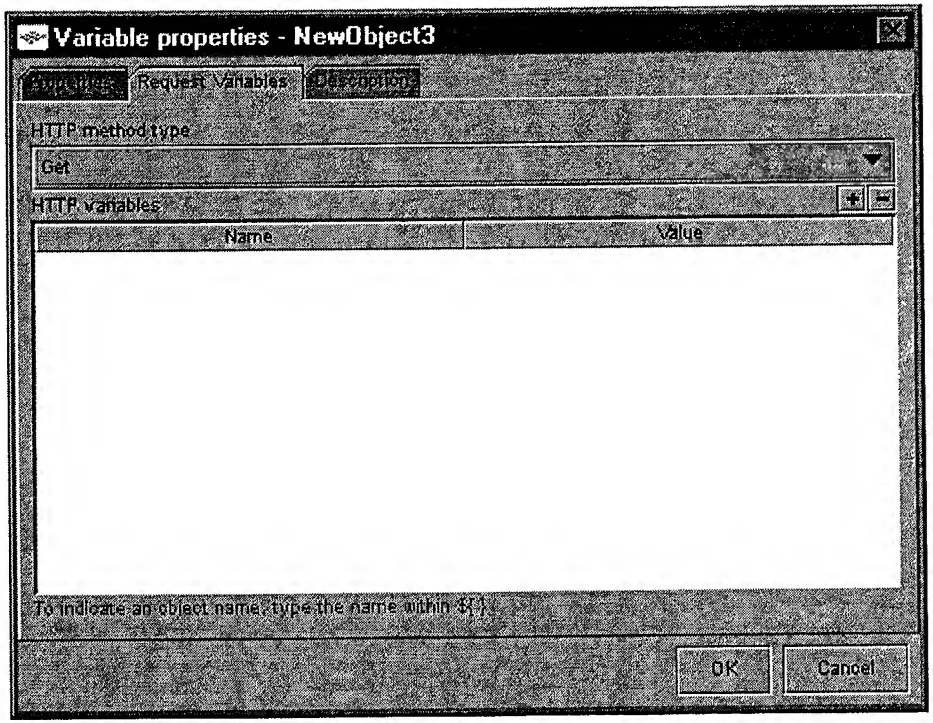
Variable properties - NewObject3
Request Variables
Description
HTTP method type
Get
HTTP variables
Name
Value
To indicate an object name, type the name within ${}.
OK
Cancel

Figure 48

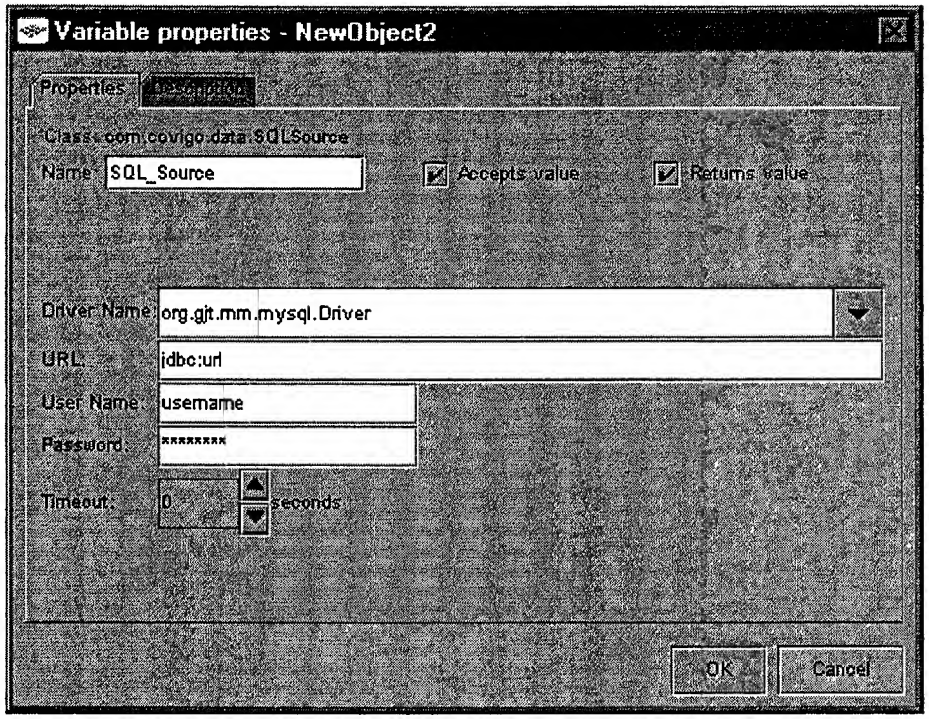
Variable properties - NewObject2
Properties
Description
Class: com.covigo.data.SQLSource
Name: SQL_Source
Accepts value
Returns value
Driver Name: org.gjt.mm.mysql.Driver
URL: idbc:url
User Name: username
Password: ********
Timeout: 0 seconds
OK
Cancel

Figure 49

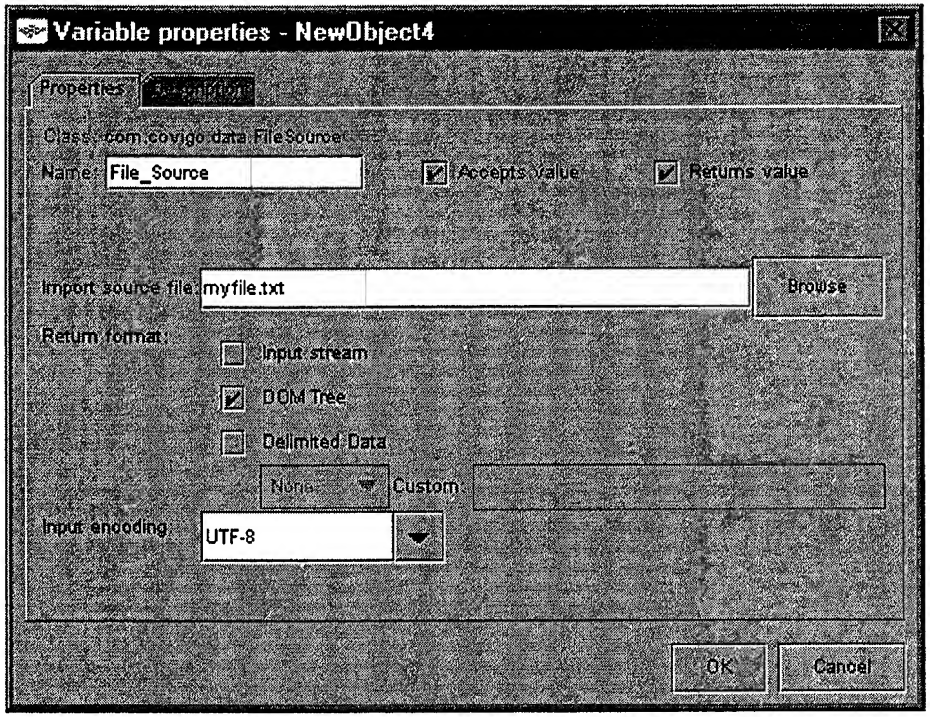
Variable properties - NewObject4
Properties
Class: com.covigo.data.FileSource
Name: File_Source
Accepts value
Returns value
Import source file: myfile.txt
Browse
Return format:
Input stream
DOM Tree
Delimited Data
None
Custom
Input encoding:
UTF-8
OK
Cancel

Figure 50

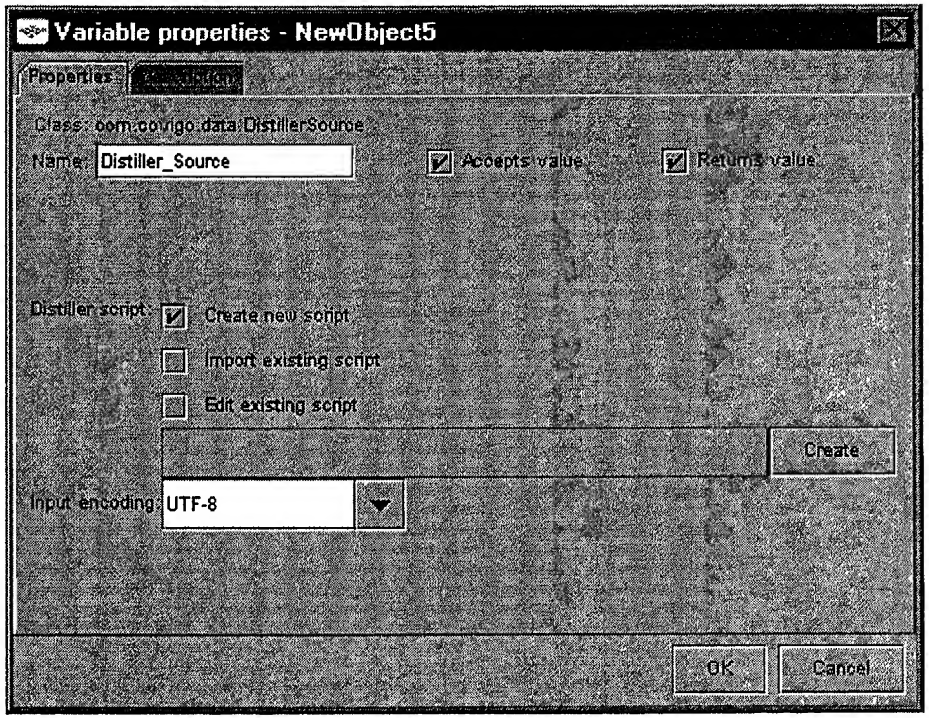
Variable properties - NewObject5
Properties
Class: com.covigo.data.DistillerSource
Name: Distiller_Source
Accepts value
Returns value
Distiller script:
Create new script
Import existing script
Edit existing script
Create
Input encoding:
UTF-8
OK
Cancel

Figure 51

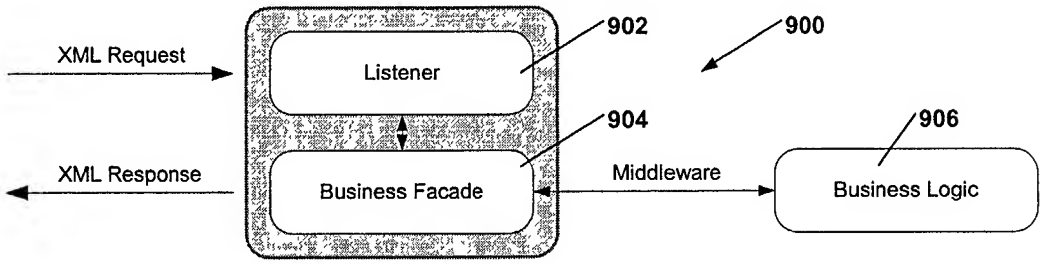
XML Request
XML Response
Listener
Business Facade
902
900
904
906
Middleware
Business Logic

Figure 52

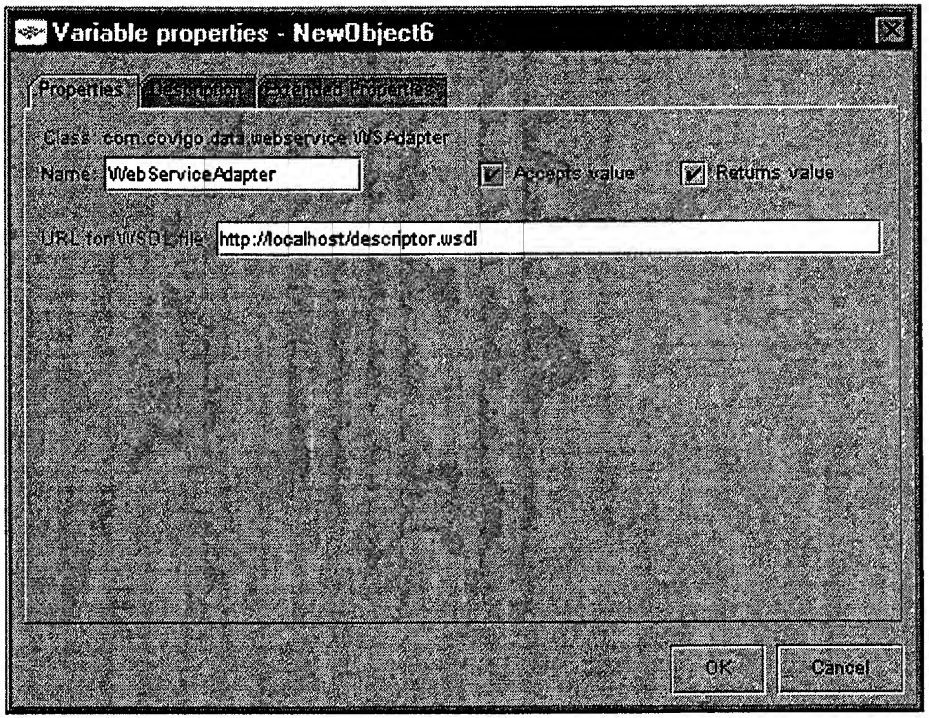
Variable properties - NewObject6
Properties
Description
Extended Properties
Class: com.covigo.data.webservice.WSAdapter
Name: WebServiceAdapter
Accepts value
Returns value
URL for WSDL file: http://localhost/descriptor.wsdl
OK
Cancel

Figure 53

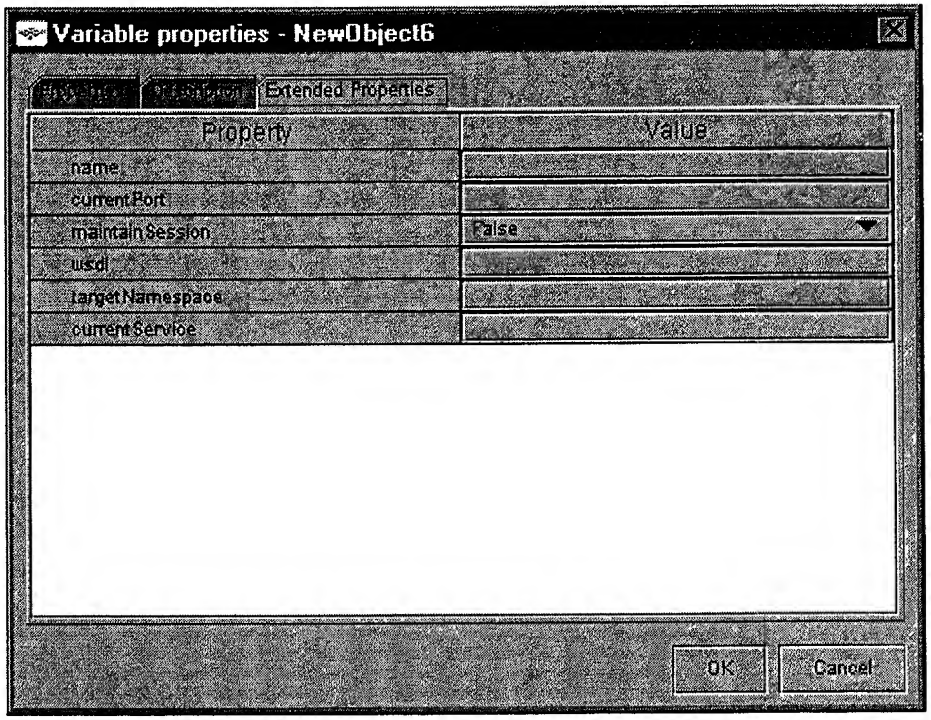
Variable properties - NewObject6
Extended Properties
Property
Value
name
currentPort
maintainSession
false
wsdl
targetNamespace
currentService
OK
Cancel

Figure 54

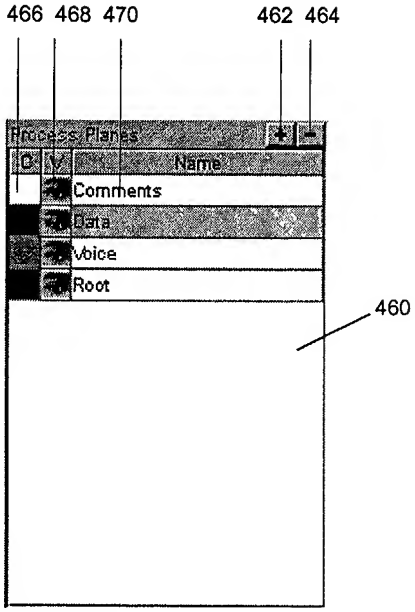
466
468
470
462
464
Process Planes
C
V
Name
Comments
Data
Voice
Root
460

Figure 55

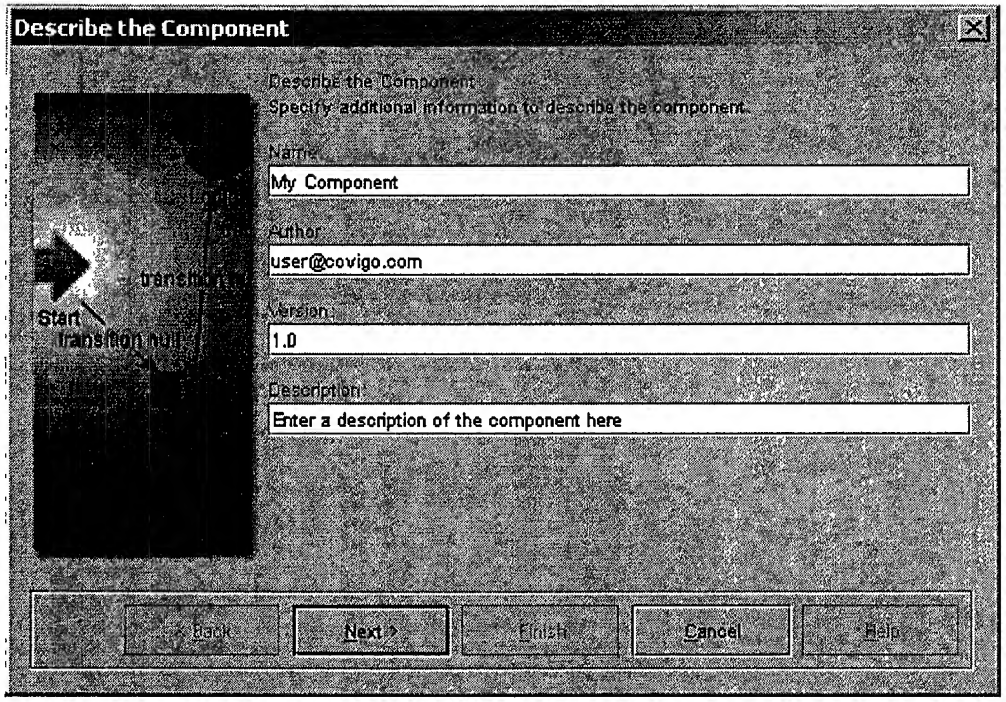
Describe the Component
Describe the Component
Specify additional information to describe the component.
Name:
My Component
Author:
user@covigo.com
Version:
1.0
Description:
Enter a description of the component here
Start
transition null
Back
Next >
Finish
Cancel
Help

Figure 56

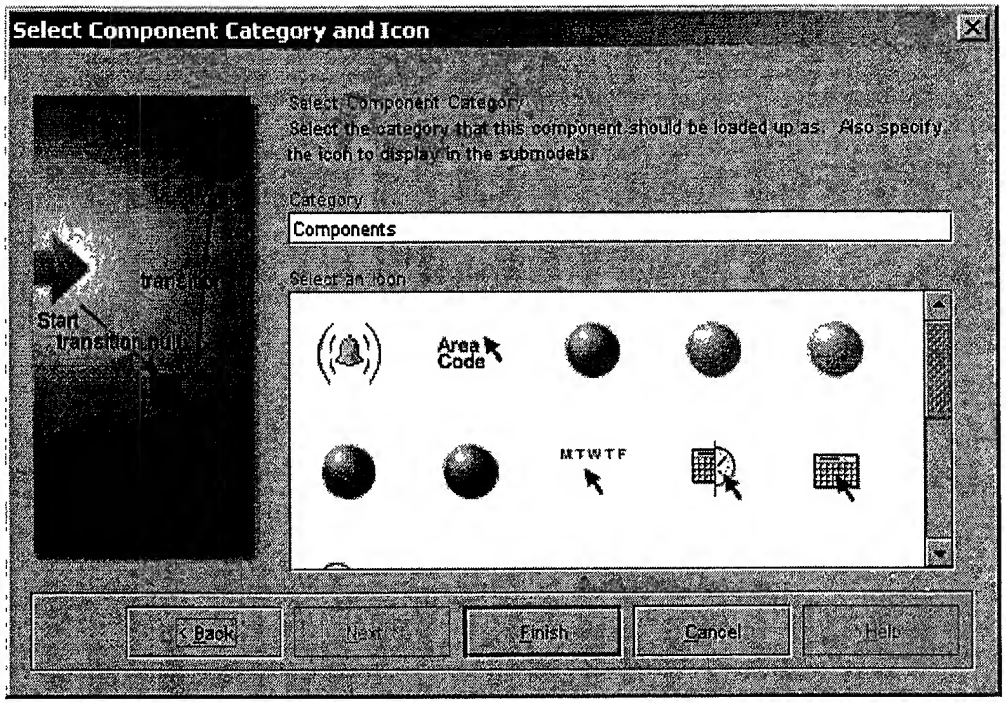
Select Component Category and Icon
Select Component Category
Select the category that this component should be loaded up as. Also specify the icon to display in the submodels.
Category
Components
Select an Icon
Start
Area Code
MTWTF
< Back
Next
Finish
Cancel
Help

Figure 57

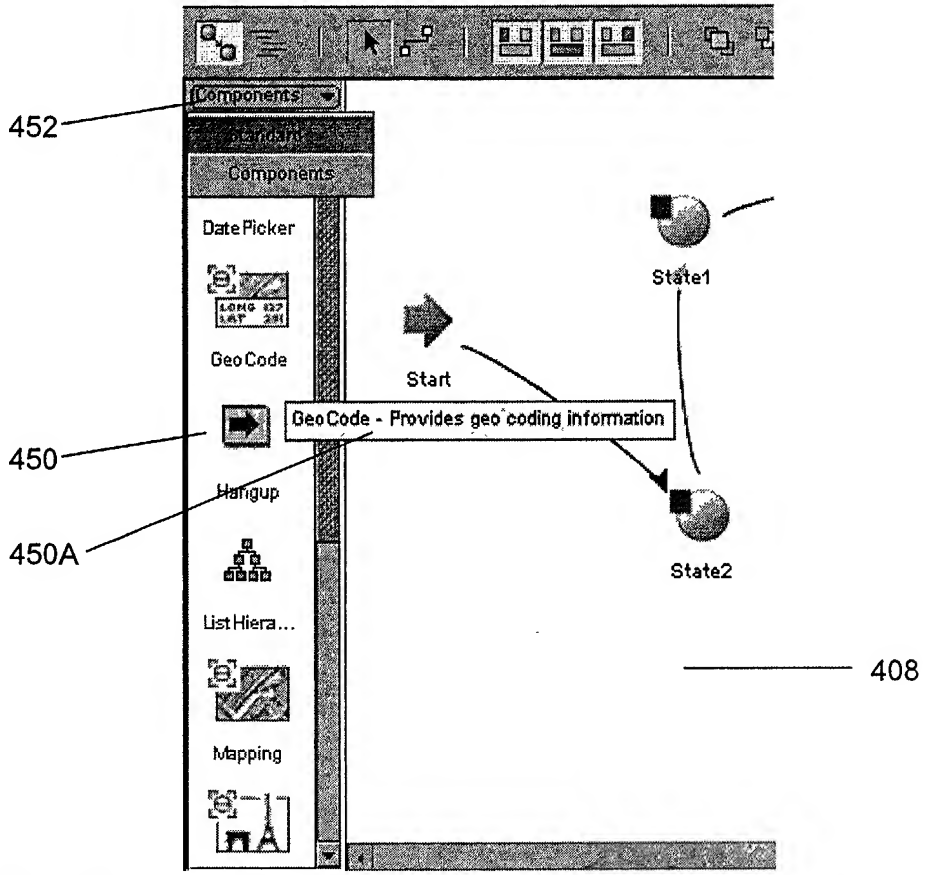
452
Components
Components
DatePicker
GeoCode
450
Hangup
450A
ListHiera...
Mapping
Start
State1
GeoCode - Provides geo coding information
State2
408

Figure 58

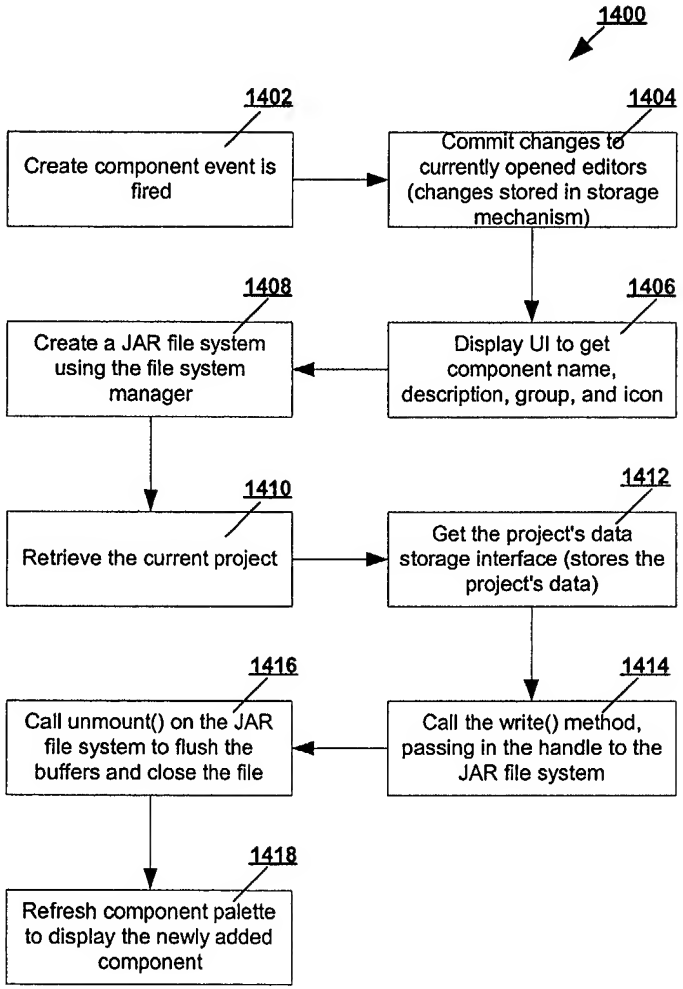
1400
1402
Create component event is fired
1404
Commit changes to currently opened editors (changes stored in storage mechanism)
1408
Create a JAR file system using the file system manager
1406
Display UI to get component name, description, group, and icon
1410
Retrieve the current project
1412
Get the project's data storage interface (stores the project's data)
1416
Call unmount() on the JAR file system to flush the buffers and close the file
1414
Call the write() method, passing in the handle to the JAR file system
1418
Refresh component palette to display the newly added component

Figure 59

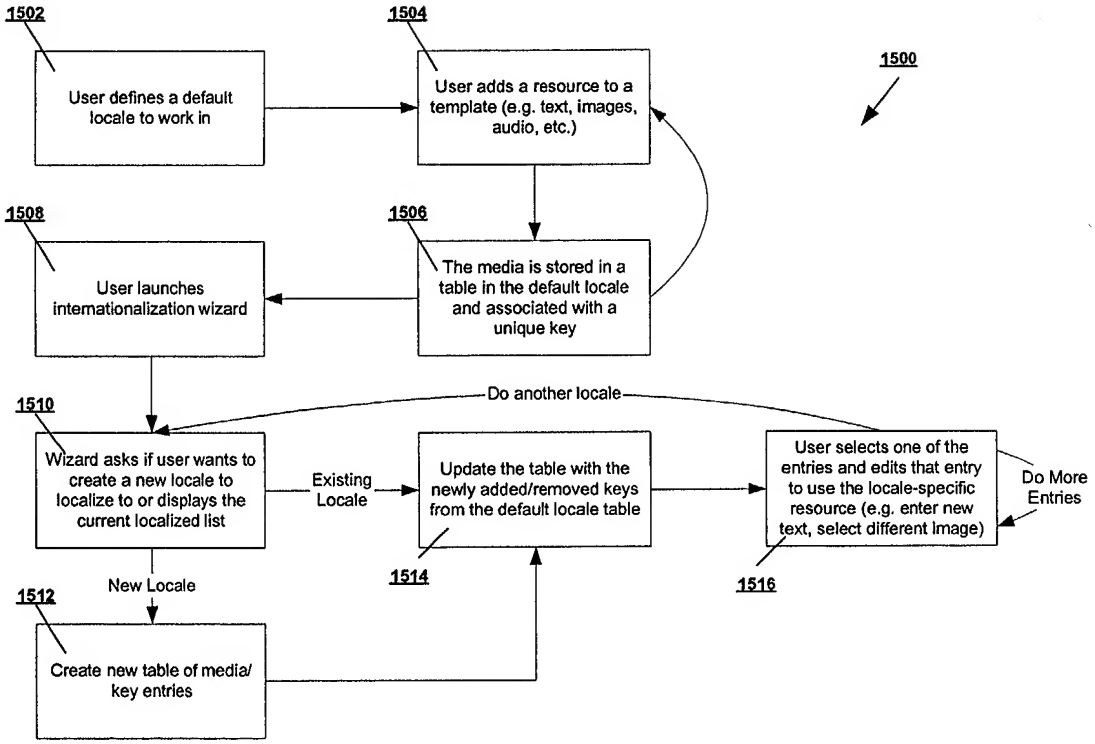
1502
User defines a default locale to work in
1504
User adds a resource to a template (e.g. text, images, audio, etc.)
1500
1508
User launches internationalization wizard
1506
The media is stored in a table in the default locale and associated with a unique key
Do another locale
1510
Wizard asks if user wants to create a new locale to localize to or displays the current localized list
Existing Locale
Update the table with the newly added/removed keys from the default locale table
User selects one of the entries and edits that entry to use the locale-specific resource (e.g. enter new text, select different Image)
Do More Entries
New Locale
1514
1516
1512
Create new table of media/ key entries

Figure 60

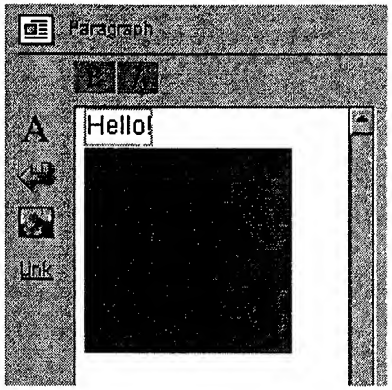

Figure 61

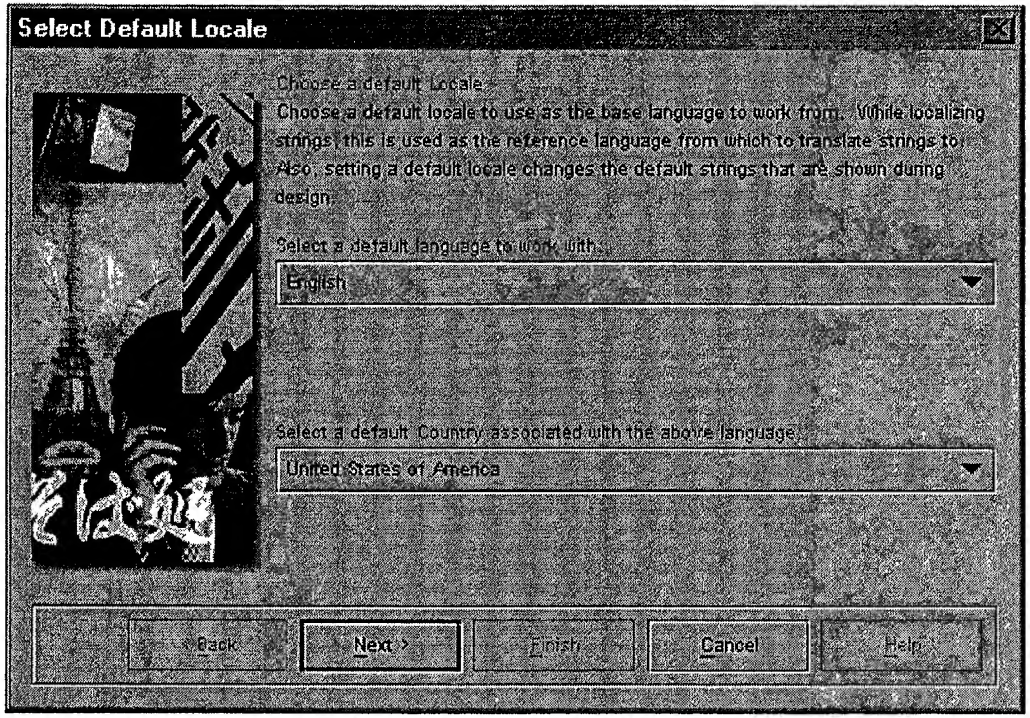
Select Default Locale
Choose a default Locale
Choose a default locale to use as the base language to work from. While localizing strings this is used as the reference language from which to translate strings to. Also, setting a default locale changes the default strings that are shown during design.
Select a default language to work with:
English
Select a default Country associated with the above language
United States of America
< Back
Next >
Finish
Cancel
Help

Figure 62

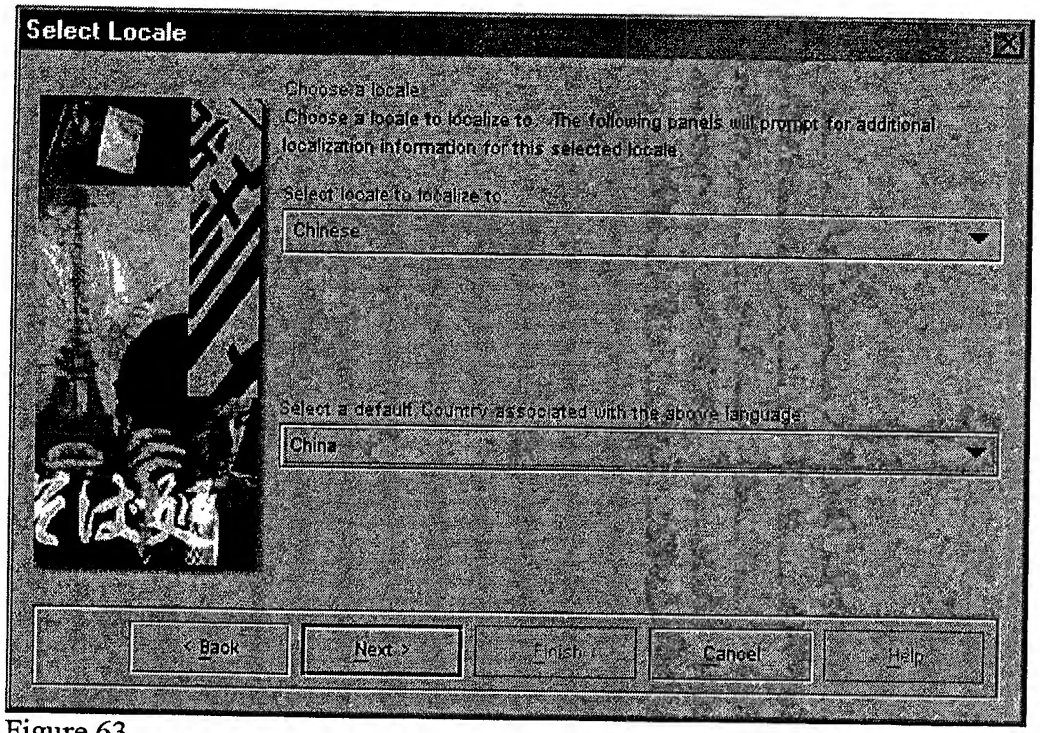
Select Locale
Choose a locale
Choose a locale to localize to. The following panels will prompt for additional localization information for this selected locale.
Select locale to localize to:
Chinese
Select a default Country associated with the above language.
China
< Back
Next >
Finish
Cancel
Help

Figure 63

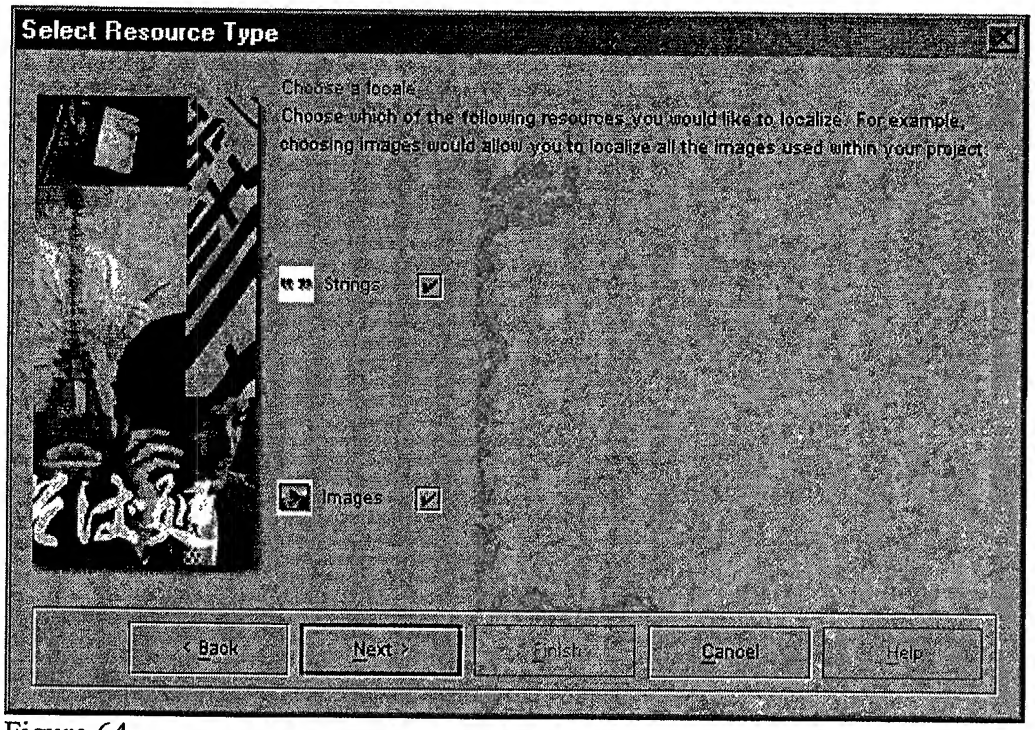
Select Resource Type
Choose a locale.
Choose which of the following resources you would like to localize. For example, choosing Images would allow you to localize all the images used within your project.
Strings
Images
< Back
Next >
Finish
Cancel
Help

Figure 64

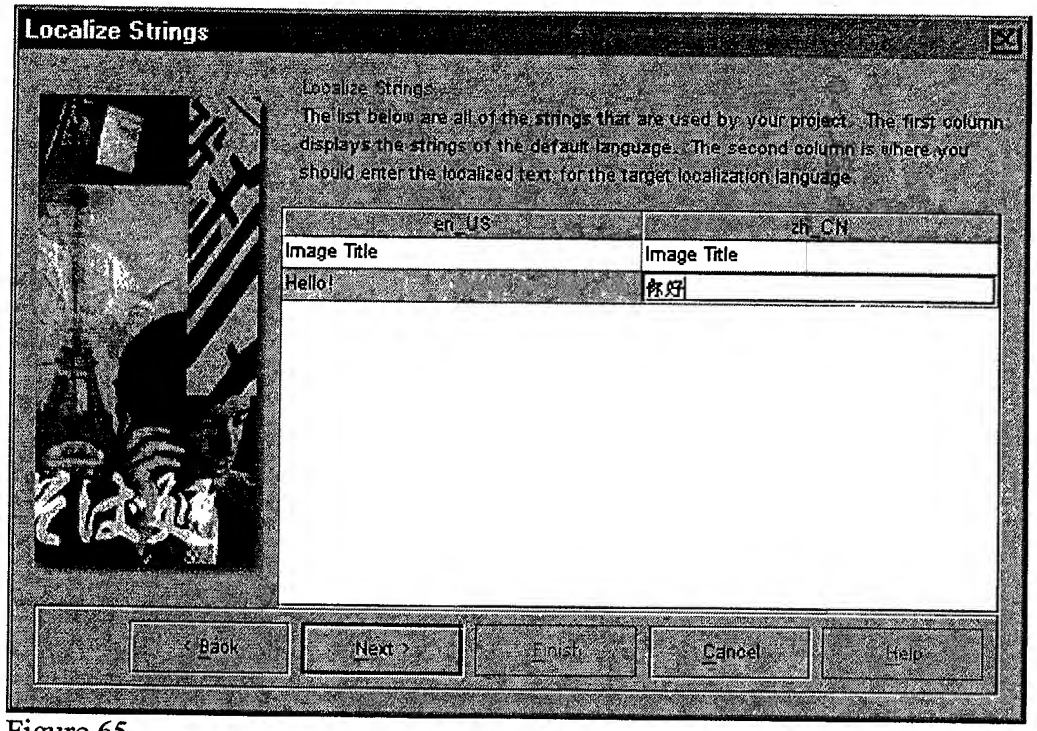
Localize Strings
Localize Strings
The list below are all of the strings that are used by your project. The first column displays the strings of the default language. The second column is where you should enter the localized text for the target localization language.
en_US
zh_CN
Image Title
Image Title
Hello!
你好
< Back
Next >
Finish
Cancel
Help

Figure 65

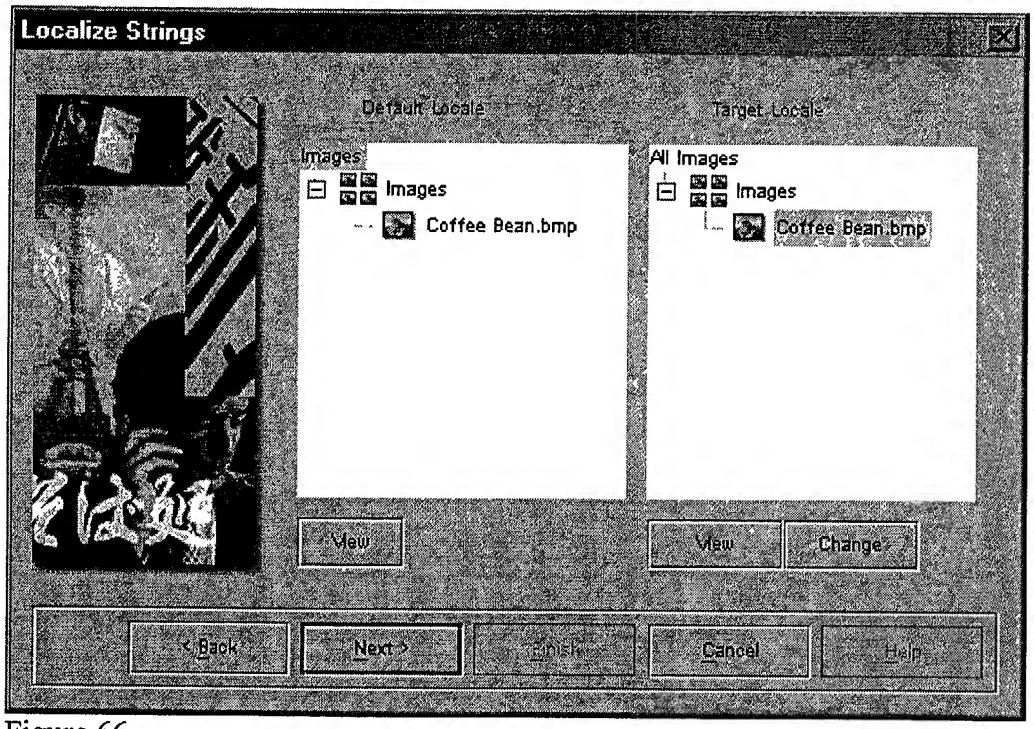
Localize Strings
Default Locale
Target Locale
Images
Images
Coffee Bean.bmp
All Images
Images
Coffee Bean.bmp
View
View
Change
< Back
Next >
Finish
Cancel
Help

Figure 66

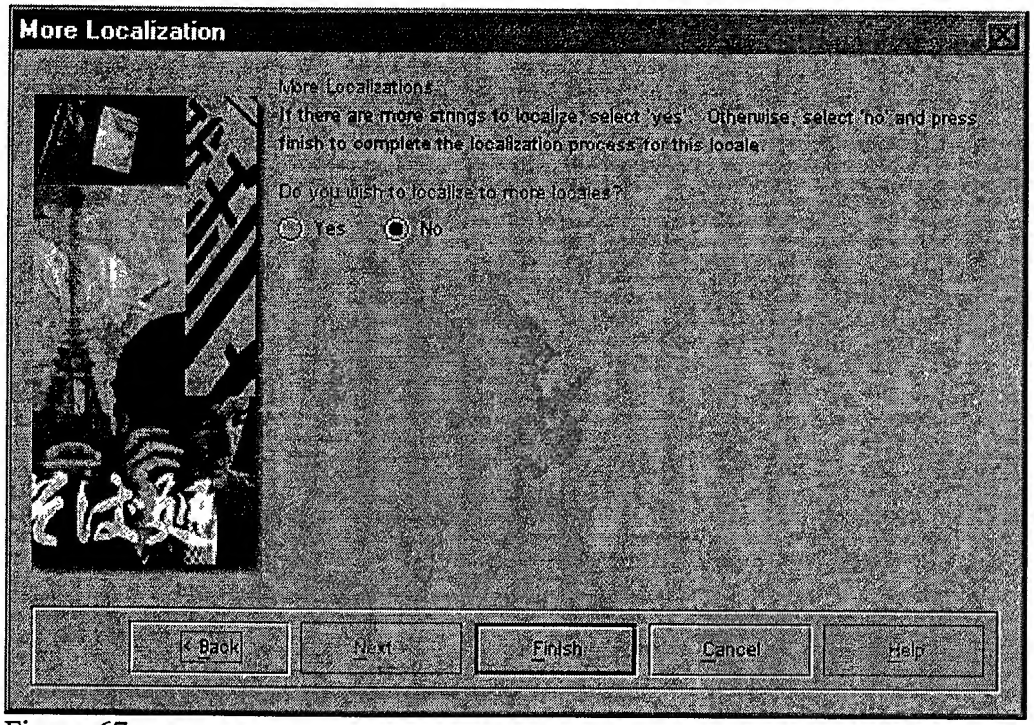

Figure 67

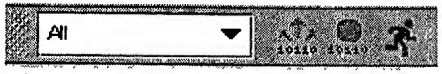

Figure 68

20070225h9001

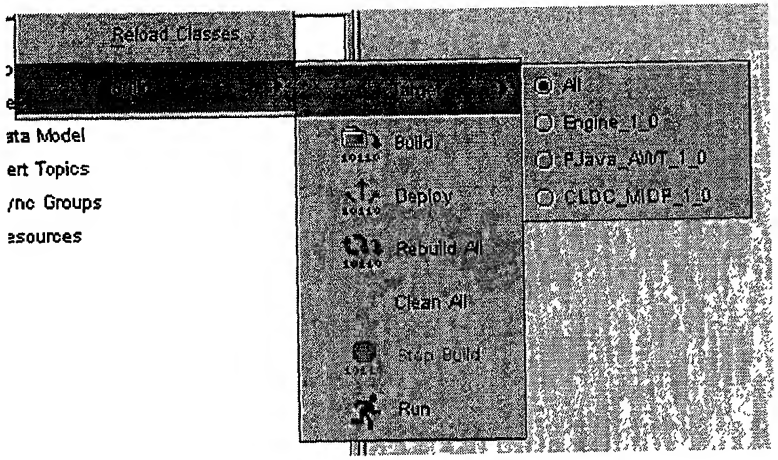

Figure 69

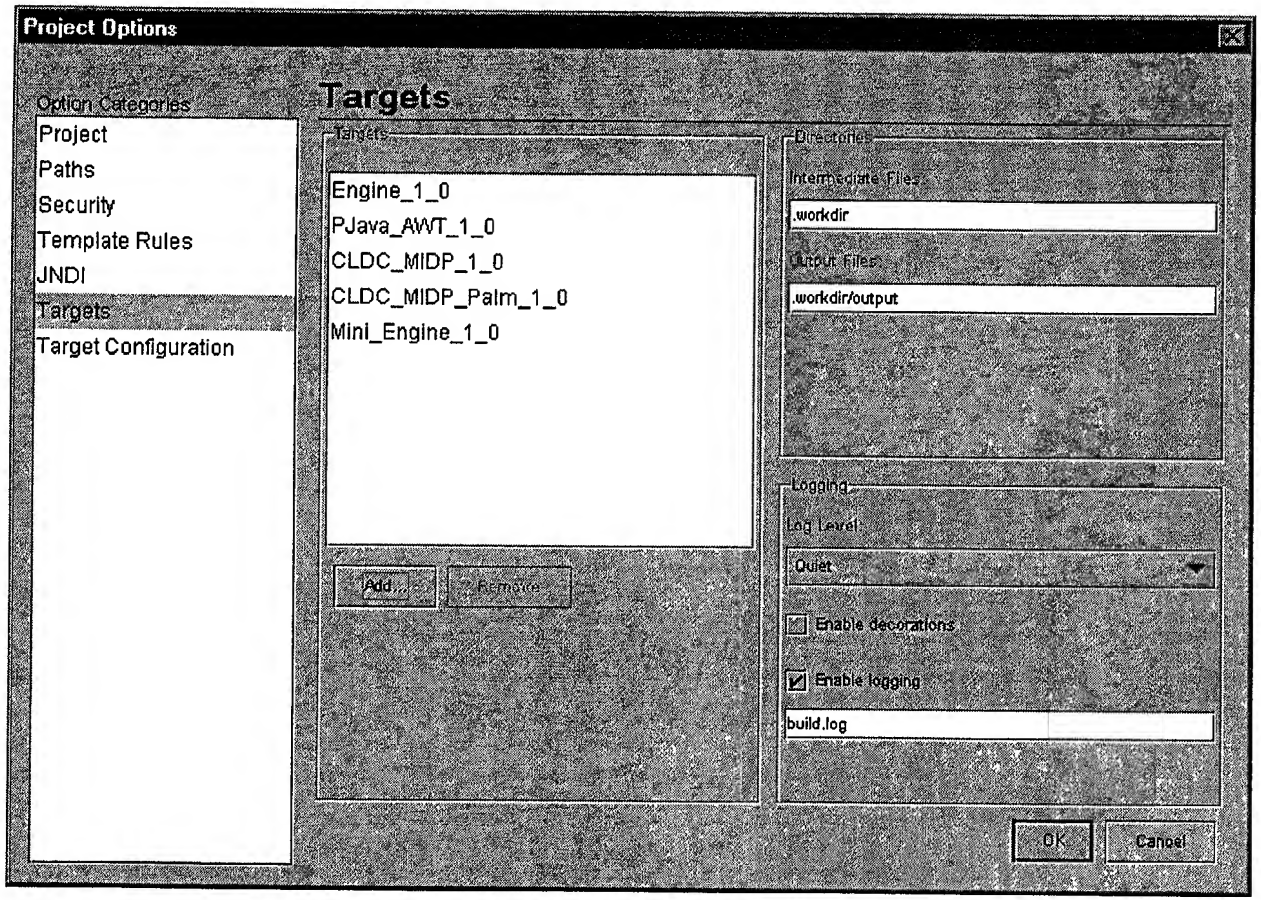
Project Options
Option Categories
Project
Paths
Security
Template Rules
JNDI
Targets
Target Configuration
Targets
Targets
Engine_1_0
PJava_AWT_1_0
CLDC_MIDP_1_0
CLDC_MIDP_Palm_1_0
Mini_Engine_1_0
Add...
Remove
Directories
Intermediate Files:
.workdir
Output Files:
.workdir/output
Logging
Log Level:
Quiet
Enable decorations
Enable logging
build.log
OK
Cancel

Figure 70

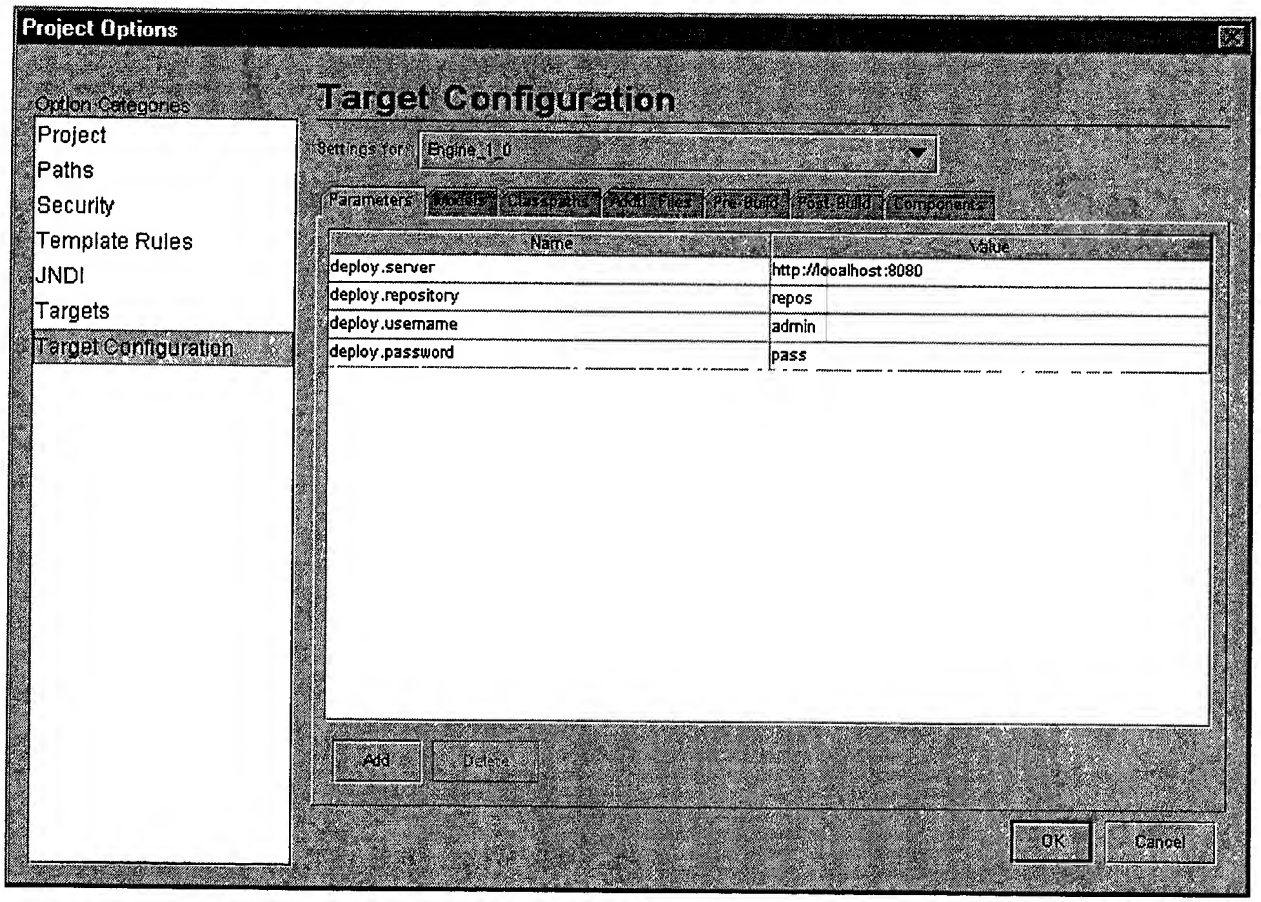
Project Options
Option Categories
Project
Paths
Security
Template Rules
JNDI
Targets
Target Configuration
Target Configuration
Settings for: Engine_1.0
Parameters
Name
Value
deploy.server
http://localhost:8080
deploy.repository
repos
deploy.username
admin
deploy.password
pass
Add
Delete
OK
Cancel

Figure 71

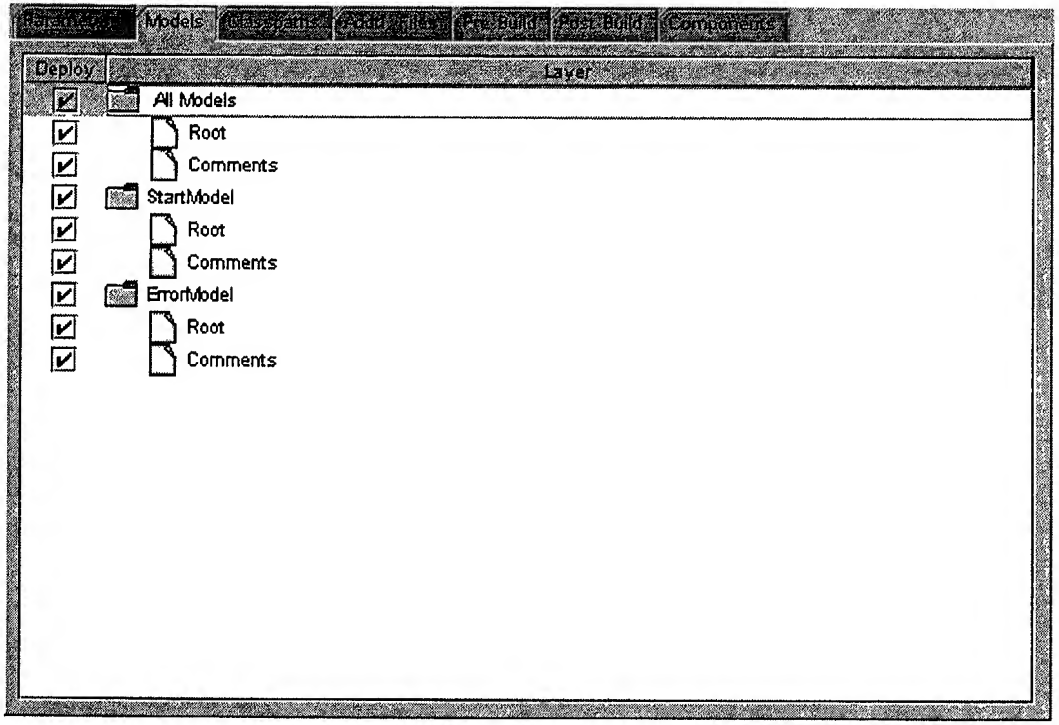
Models
Deploy
Layer
All Models
Root
Comments
StartModel
Root
Comments
ErrorModel
Root
Comments

Figure 72

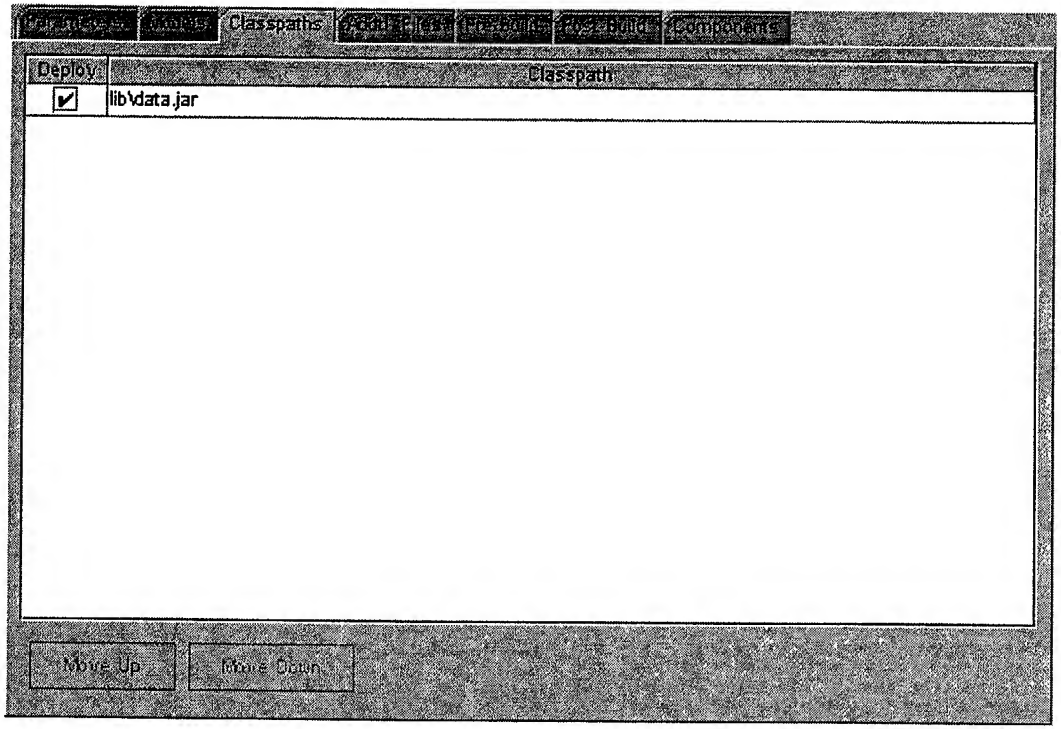

Figure 73

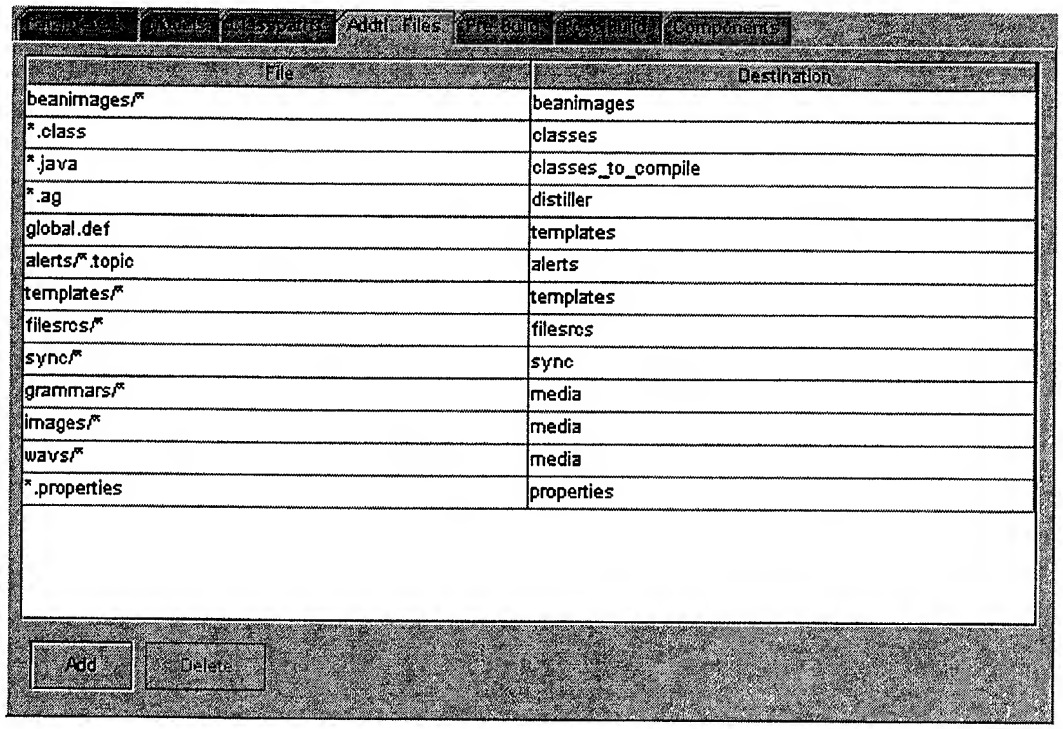
Addl. Files
File
Destination
beanimages/*
beanimages
*.class
classes
*.java
classes_to_compile
*.ag
distiller
global.def
templates
alerts/*.topic
alerts
templates/*
templates
filesres/*
filesres
sync/*
sync
grammars/*
media
images/*
media
wavs/*
media
*.properties
properties
Add
Delete

Figure 74

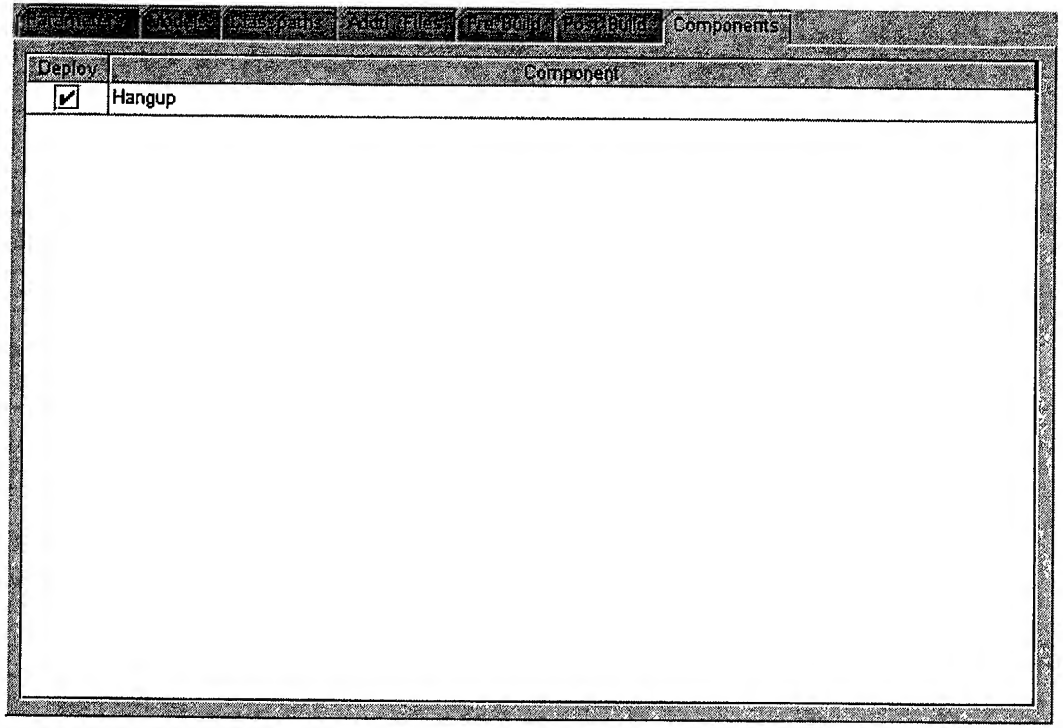

Figure 75

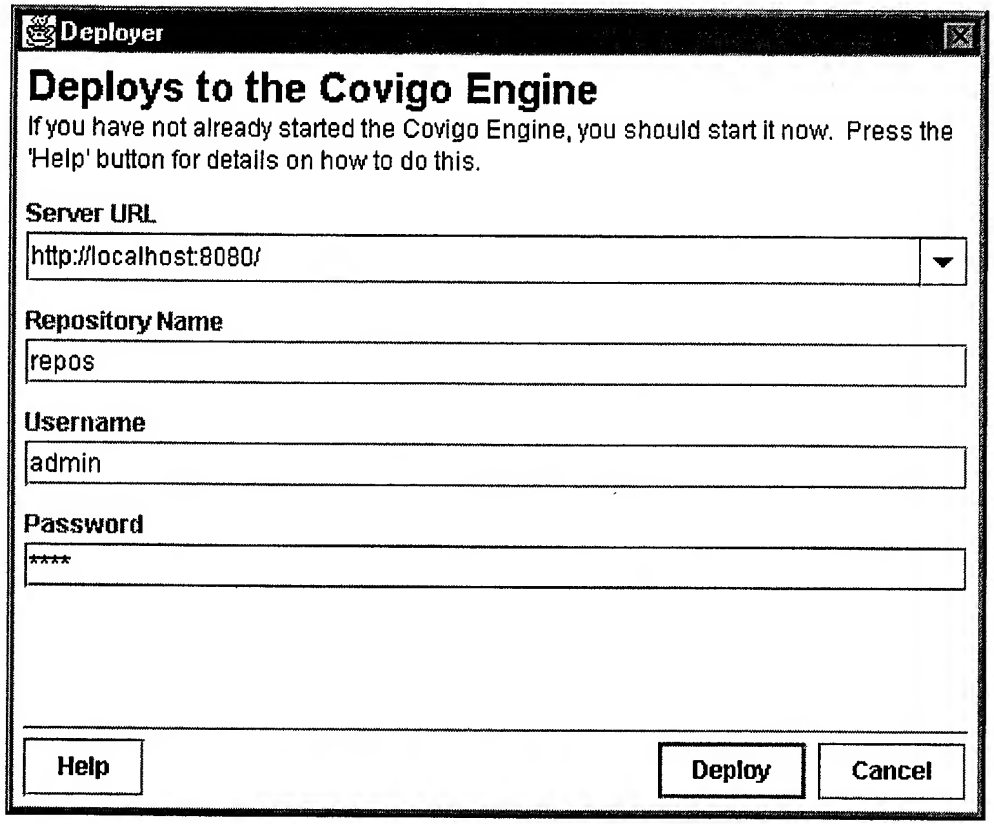
Deployer
Deploys to the Covigo Engine
If you have not already started the Covigo Engine, you should start it now. Press the 'Help' button for details on how to do this.
Server URL
http://localhost:8080/
Repository Name
repos
Username
admin
Password
****
Help
Deploy
Cancel

Figure 76

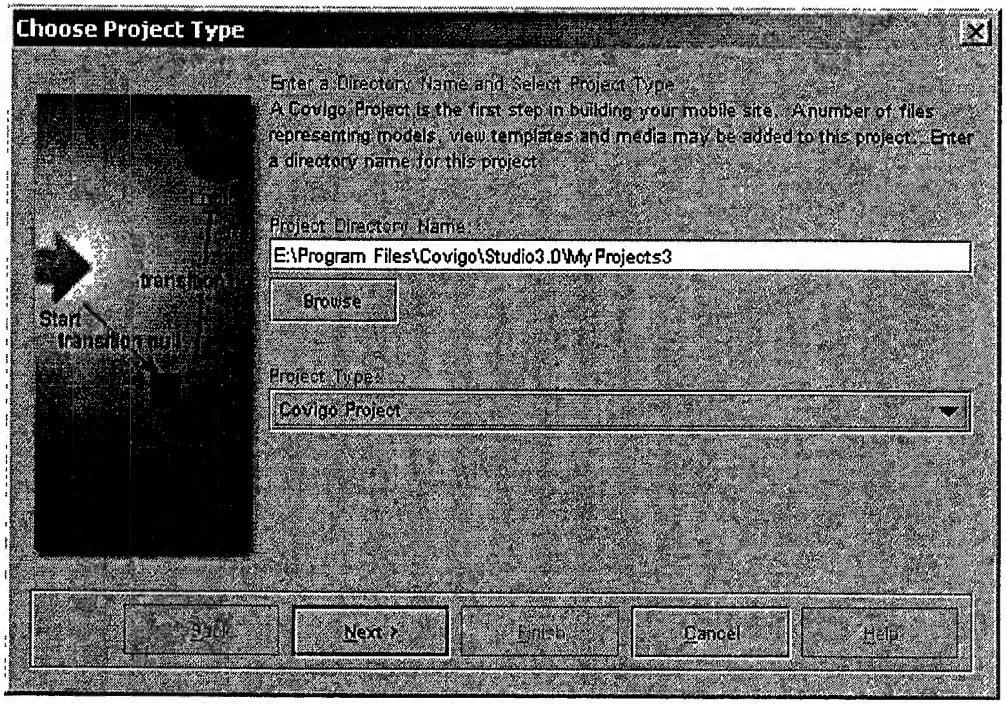
Choose Project Type
Enter a Directory Name and Select Project Type
A Covigo Project is the first step in building your mobile site. A number of files representing models, view templates and media may be added to this project. Enter a directory name for this project.
Project Directory Name:
E:\Program Files\Covigo\Studio3.0\My Projects3
Browse
Project Type:
Covigo Project
< Back
Next >
Finish
Cancel
Help

Figure 77

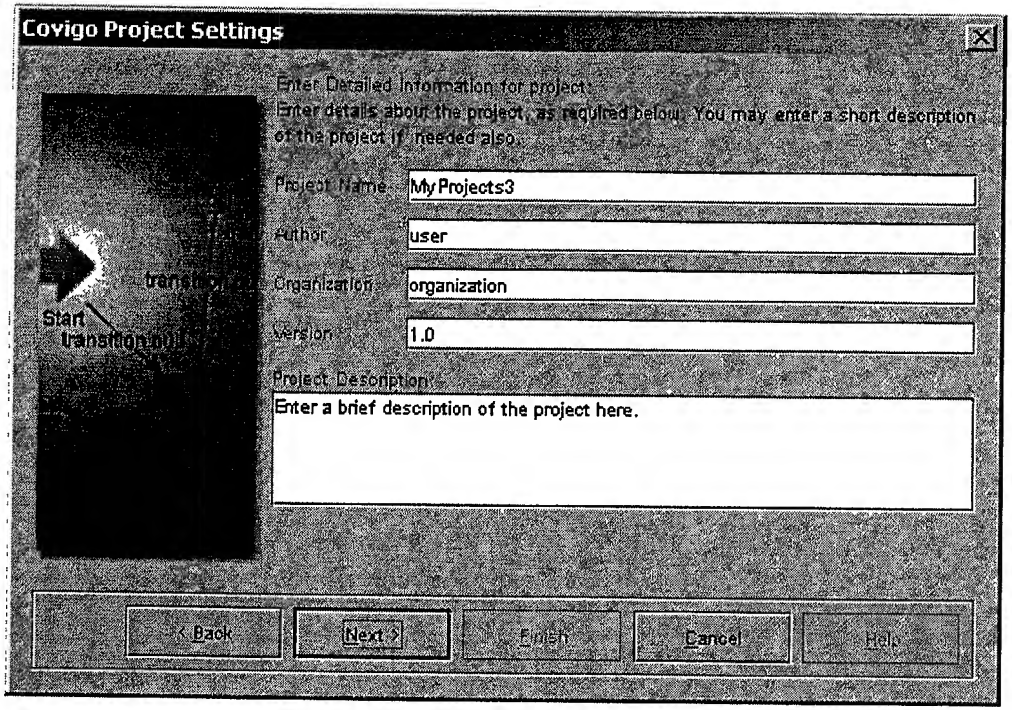
Covigo Project Settings
Enter Detailed Information for project.
Enter details about the project, as required below. You may enter a short description of the project if needed also.
Project Name:
MyProjects3
Author:
user
Organization:
organization
Version:
1.0
Project Description:
Enter a brief description of the project here.
< Back
Next >
Finish
Cancel
Help

Figure 78

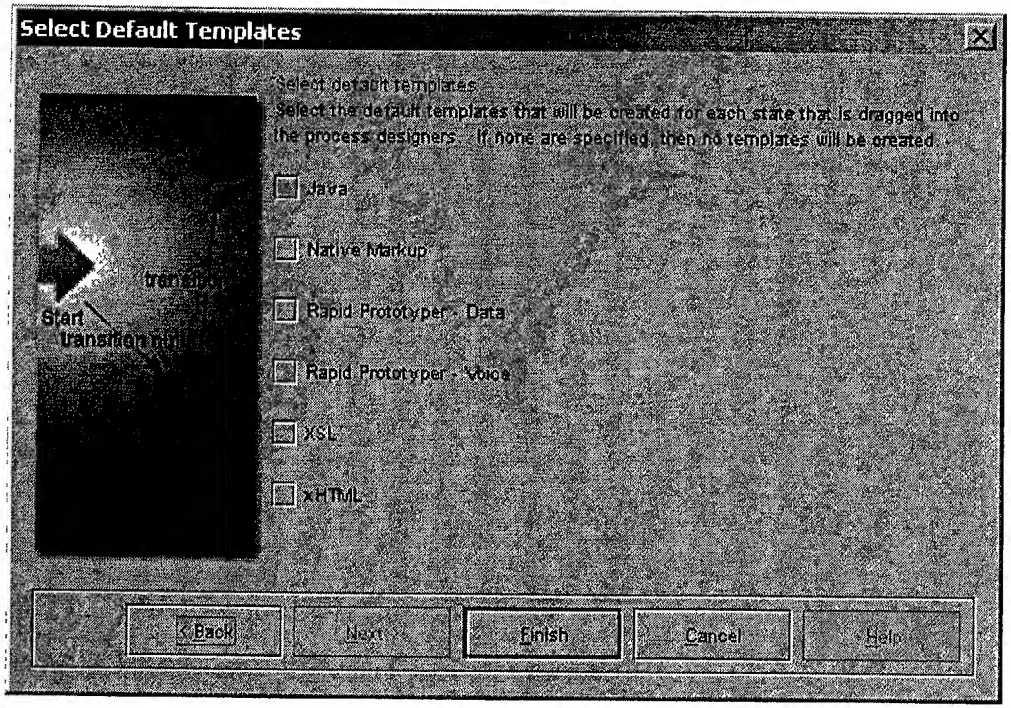
Select Default Templates
Select default templates
Select the default templates that will be created for each state that is dragged into the process designers. If none are specified, then no templates will be created.
Java
Native Markup
Rapid Prototyper - Data
Rapid Prototyper - Voice
XSL
xHTML
Start
< Back
Next >
Finish
Cancel
Help

Figure 79

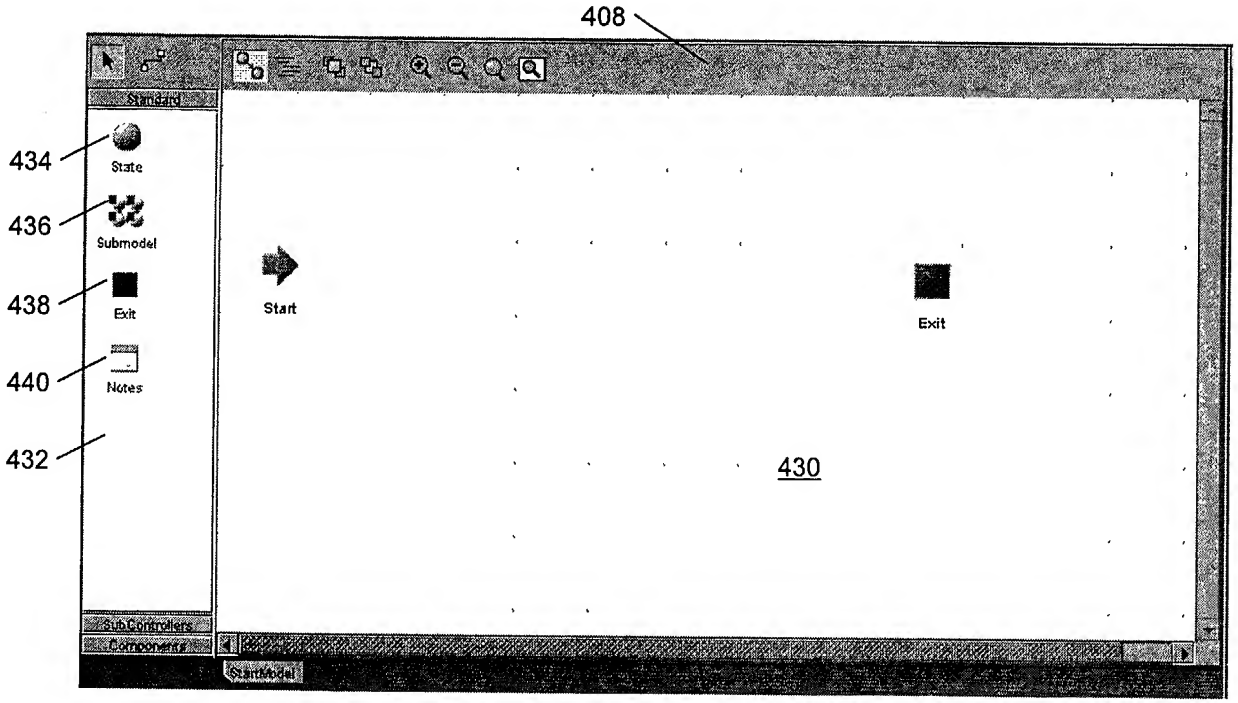
408
434
436
438
440
432
State
Submodel
Exit
Notes
Start
Exit
430
SubControllers
Components
StartModel

Figure 80

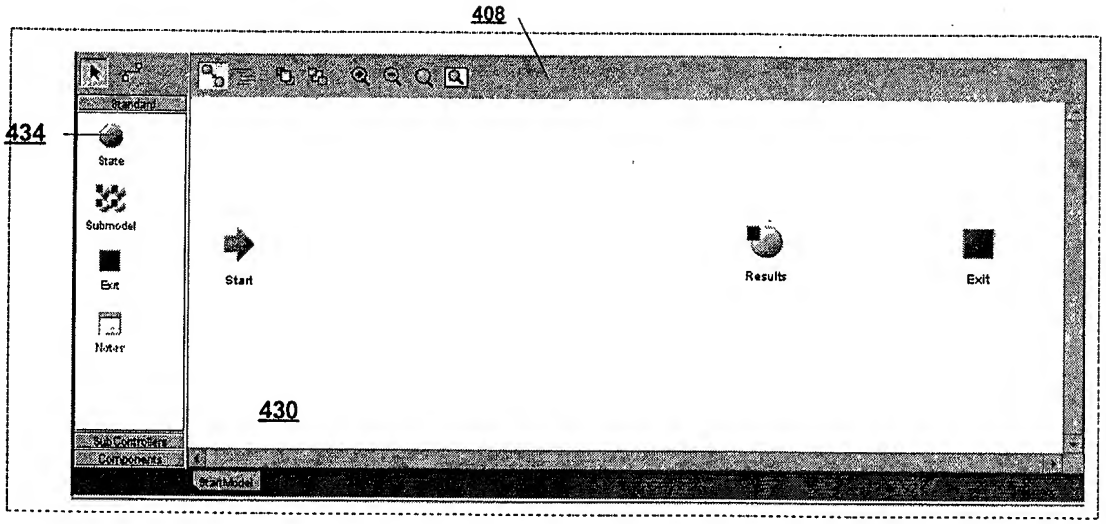

Figure 81

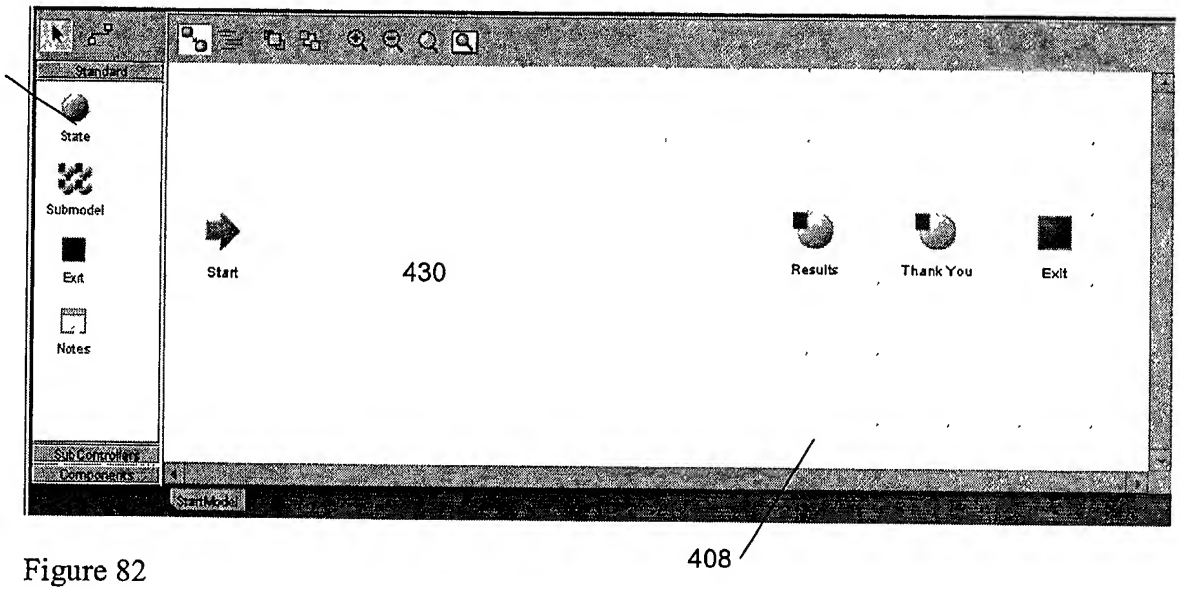
434
Standard
State
Submodel
Exit
Notes
Start
430
Results
Thank You
Exit
408

Figure 82

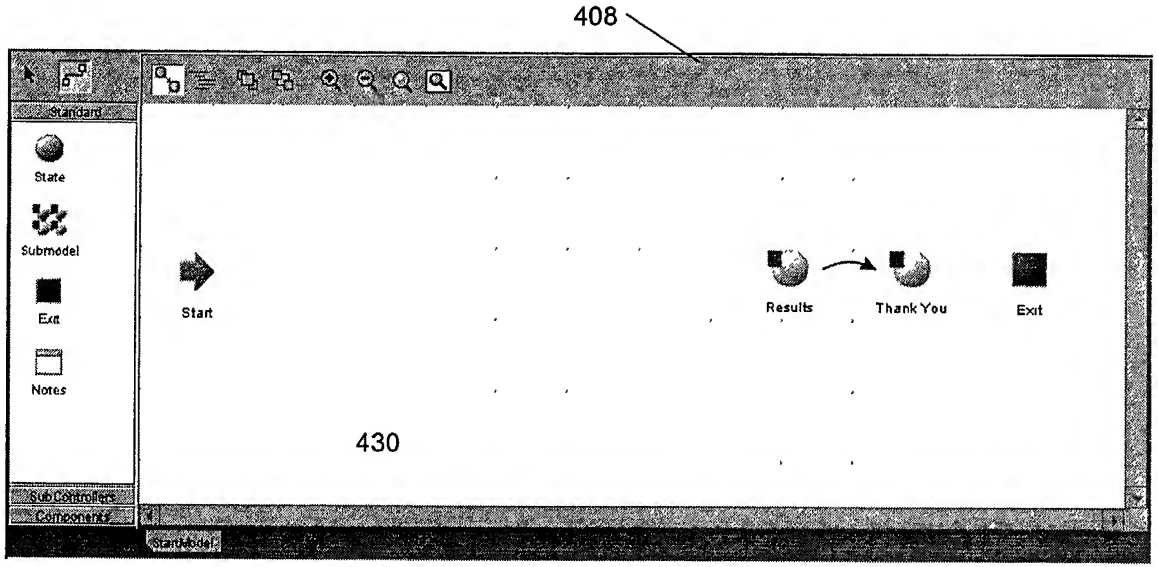
408
Standard
State
Submodel
Exit
Notes
Sub Controllers
Components
Start
Results
Thank You
Exit
430

Figure 83

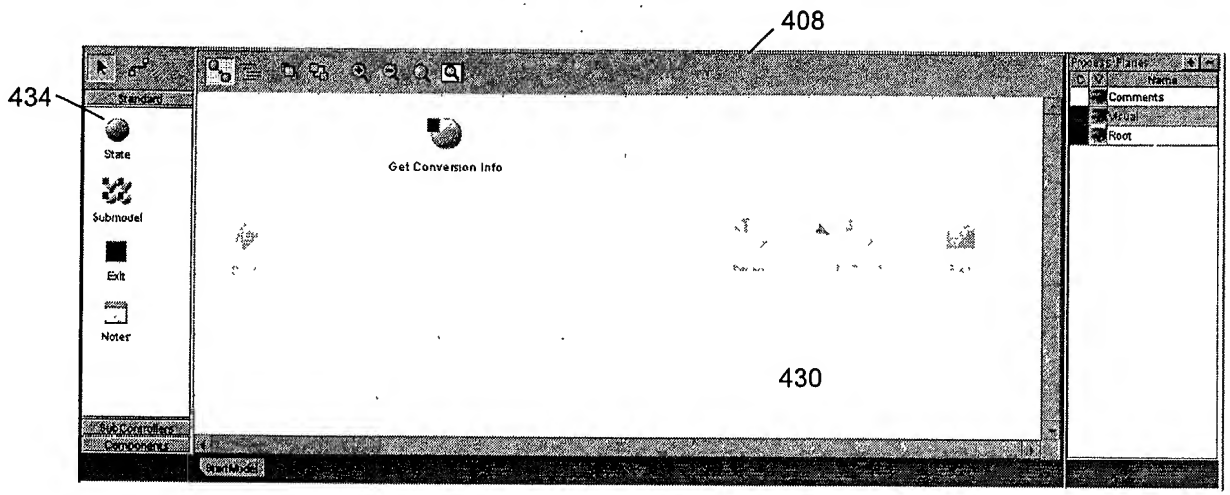

Figure 84

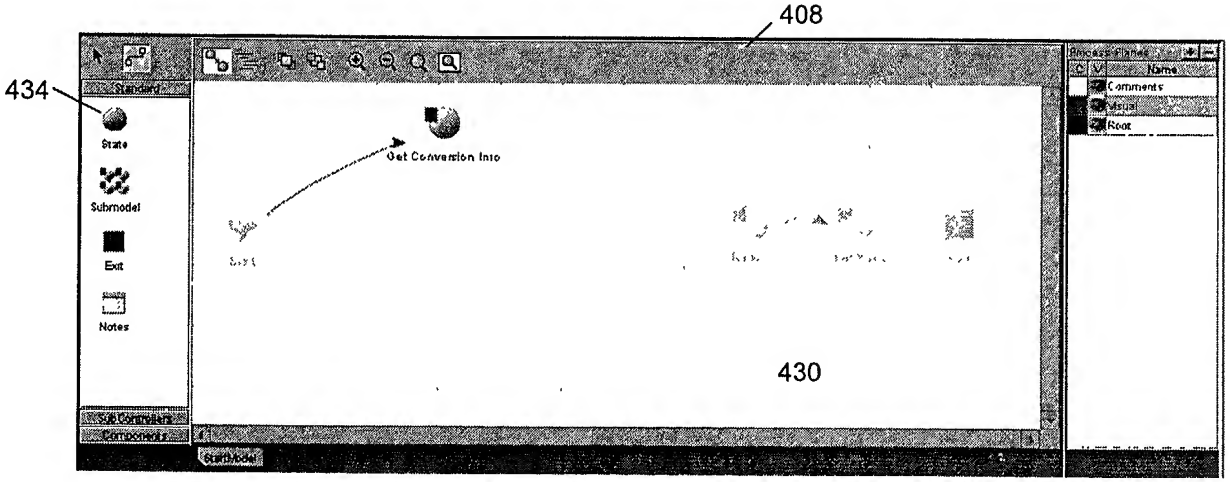

Figure 85

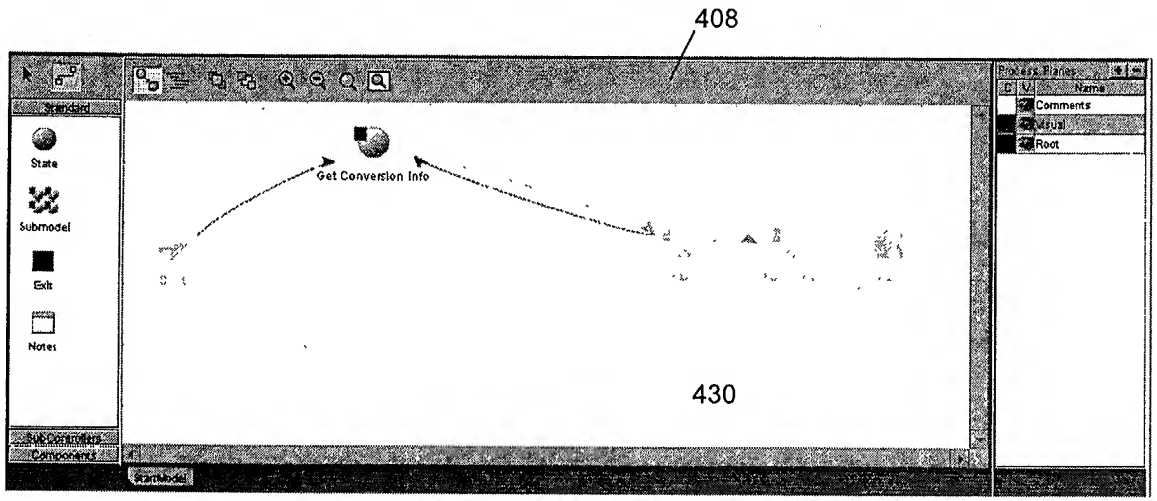

Figure 86

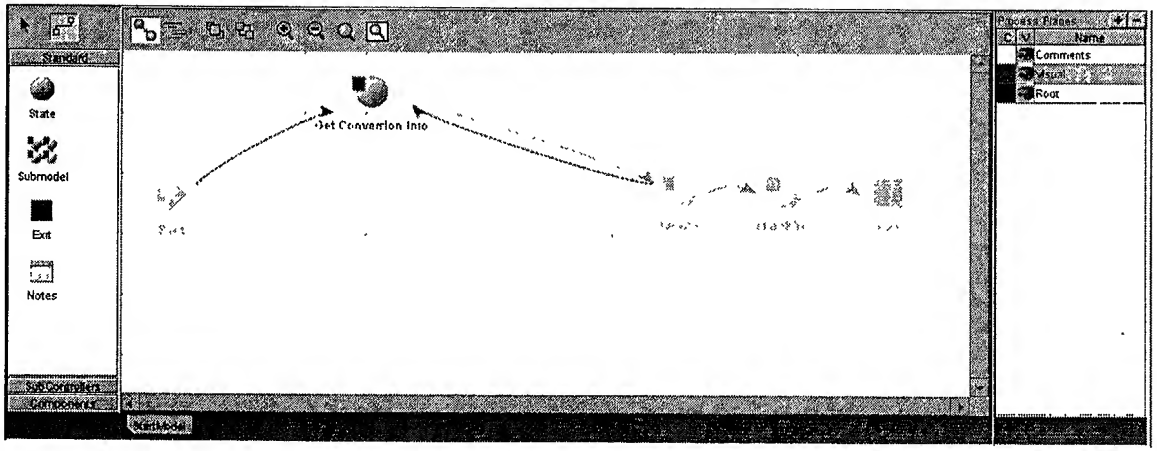

Figure 87

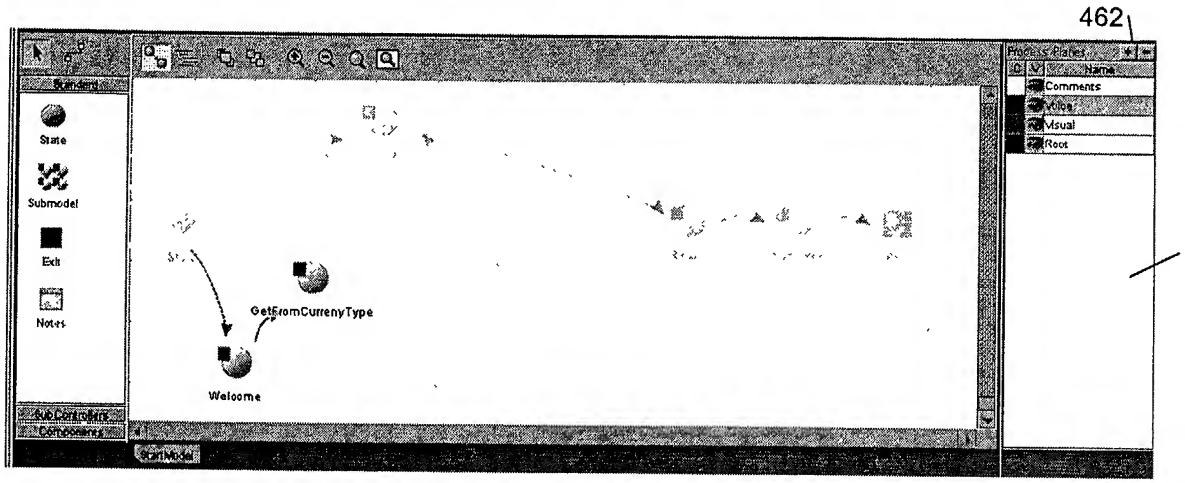

Figure 88

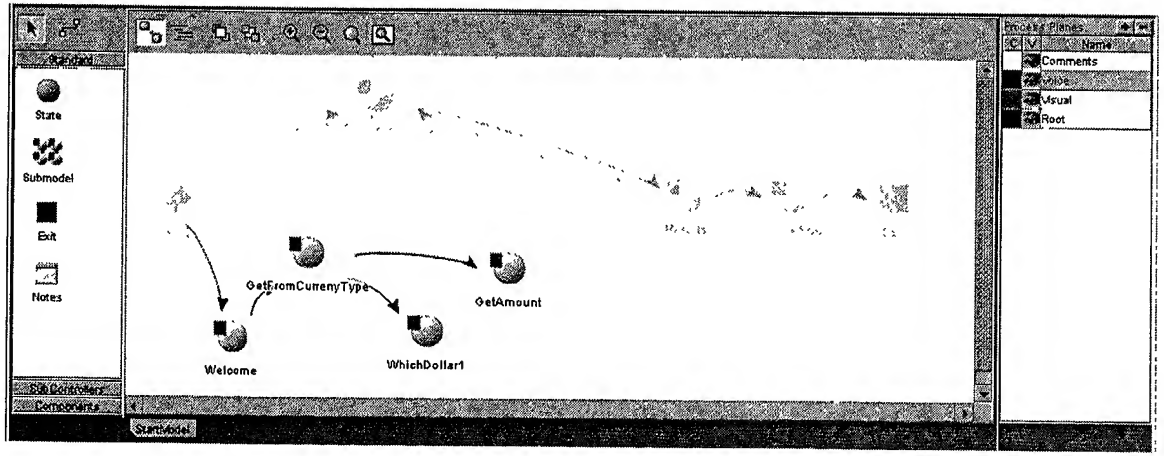

Figure 89

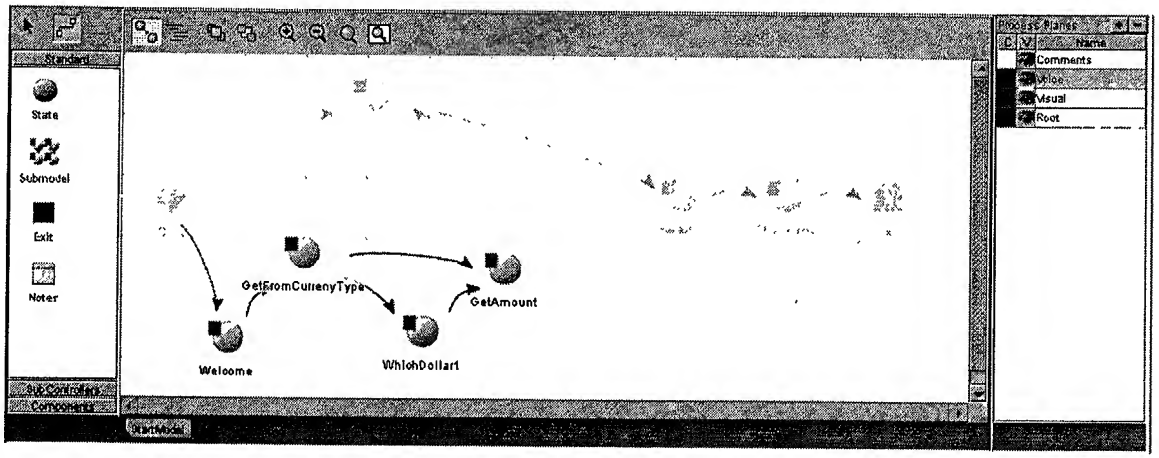

Figure 90

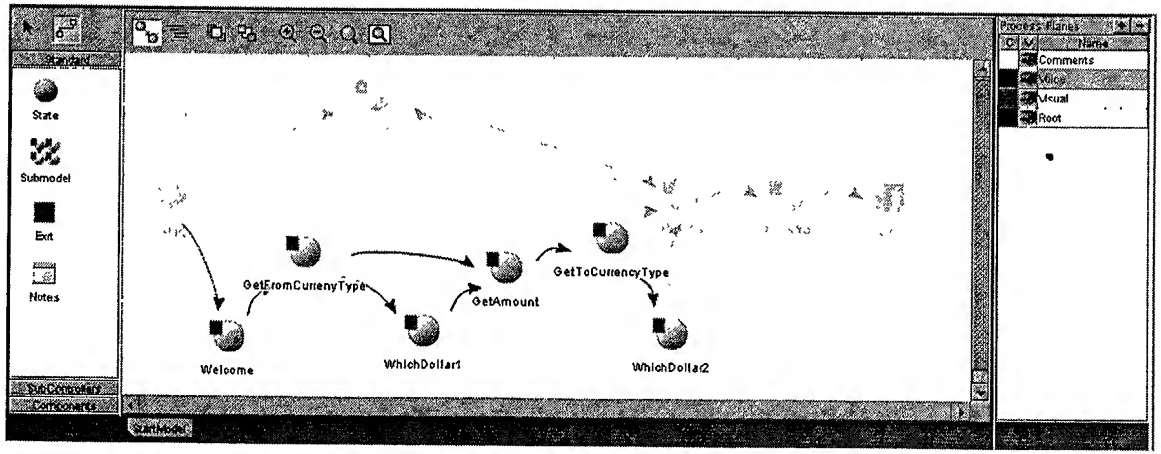

Figure 91

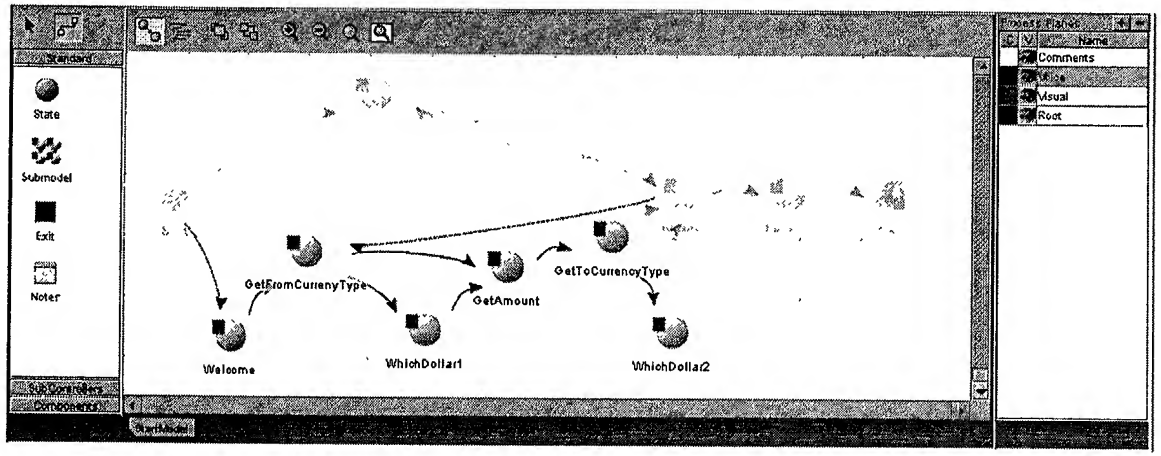

Figure 92

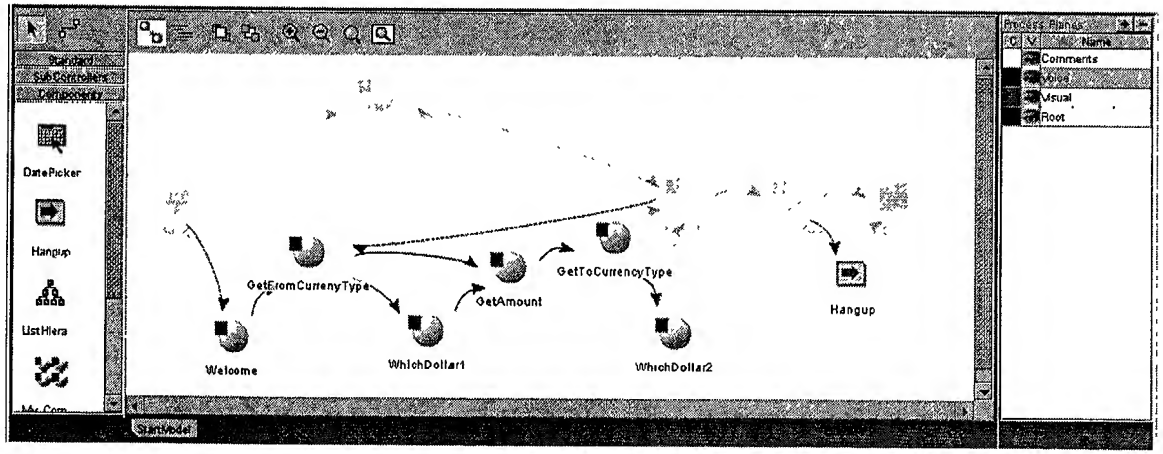

Figure 93

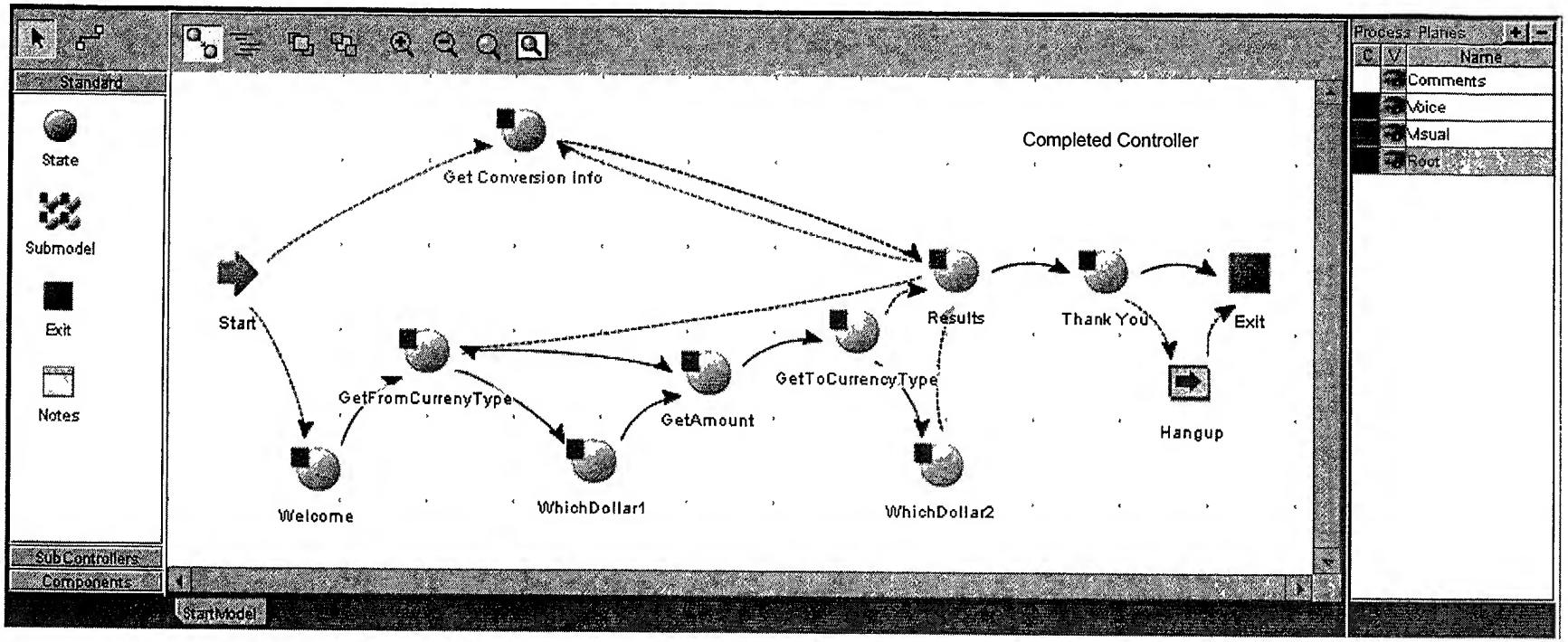
Standard
State
Submodel
Exit
Notes
Sub Controllers
Components
StartModel
Completed Controller
Get Conversion Info
Start
Results
Thank You
Exit
GetFromCurrenyType
GetAmount
GetToCurrencyType
Hangup
Welcome
WhichDollar1
WhichDollar2
Process Planes
C
V
Name
Comments
Voice
Visual
Root

Figure 94

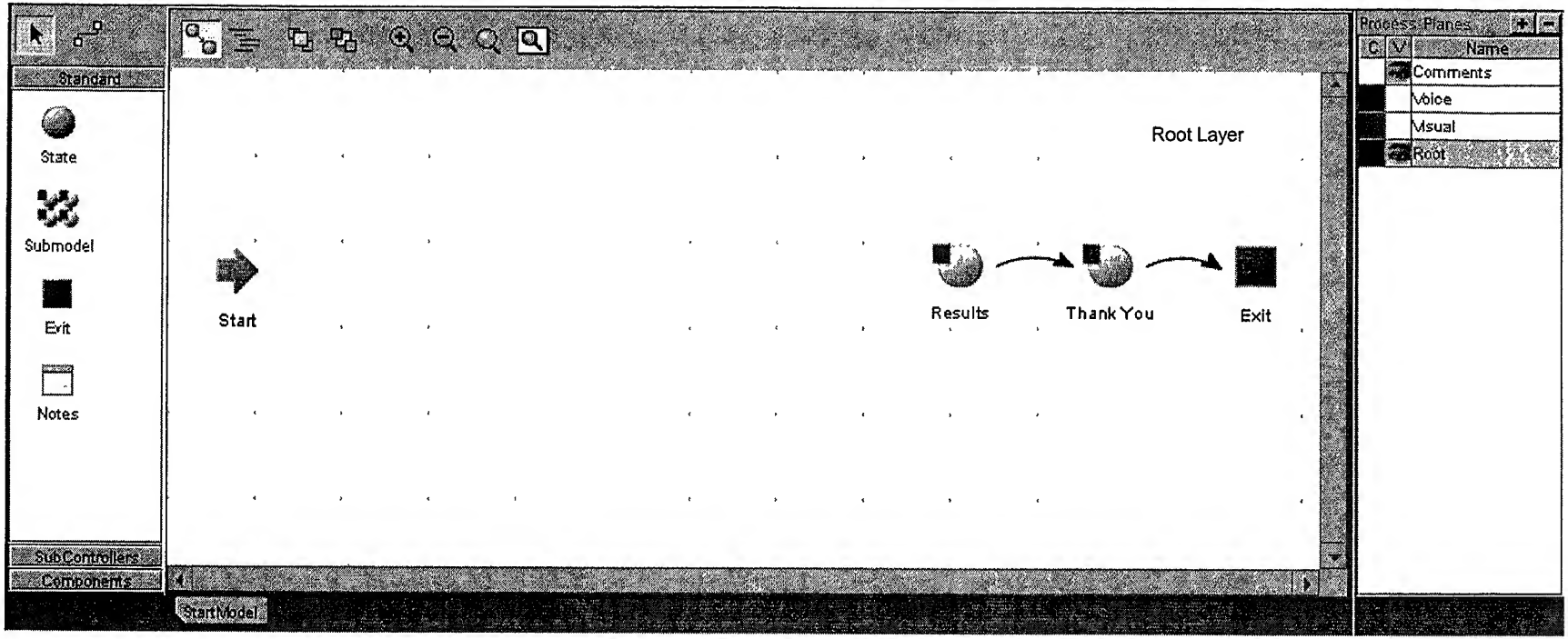
Standard
State
Submodel
Exit
Notes
SubControllers
Components
Root Layer
Start
Results
Thank You
Exit
StartModel
Process Planes
C
V
Name
Comments
Voice
Visual
Root

Figure 95

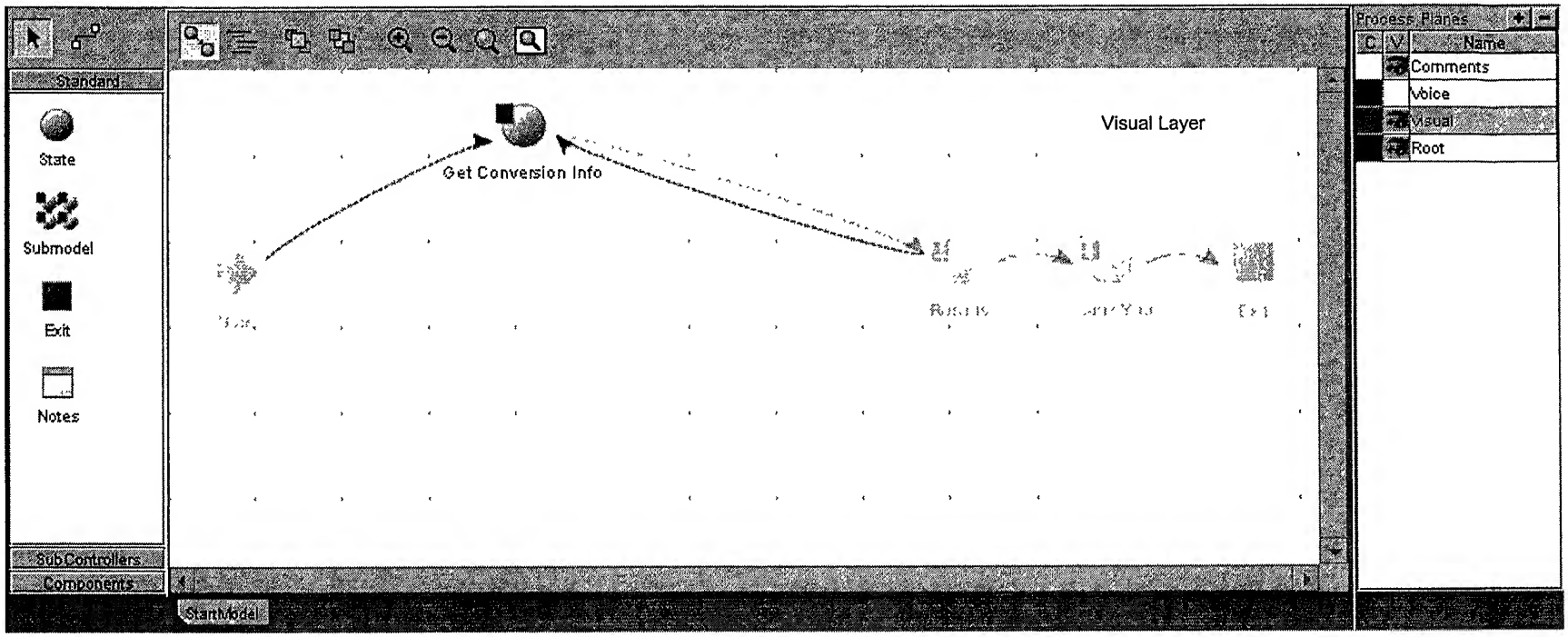
Standard
State
Submodel
Exit
Notes
SubControllers
Components
StartModel
Get Conversion Info
Visual Layer
Process Planes
C
V
Name
Comments
Voice
Visual
Root

Figure 96

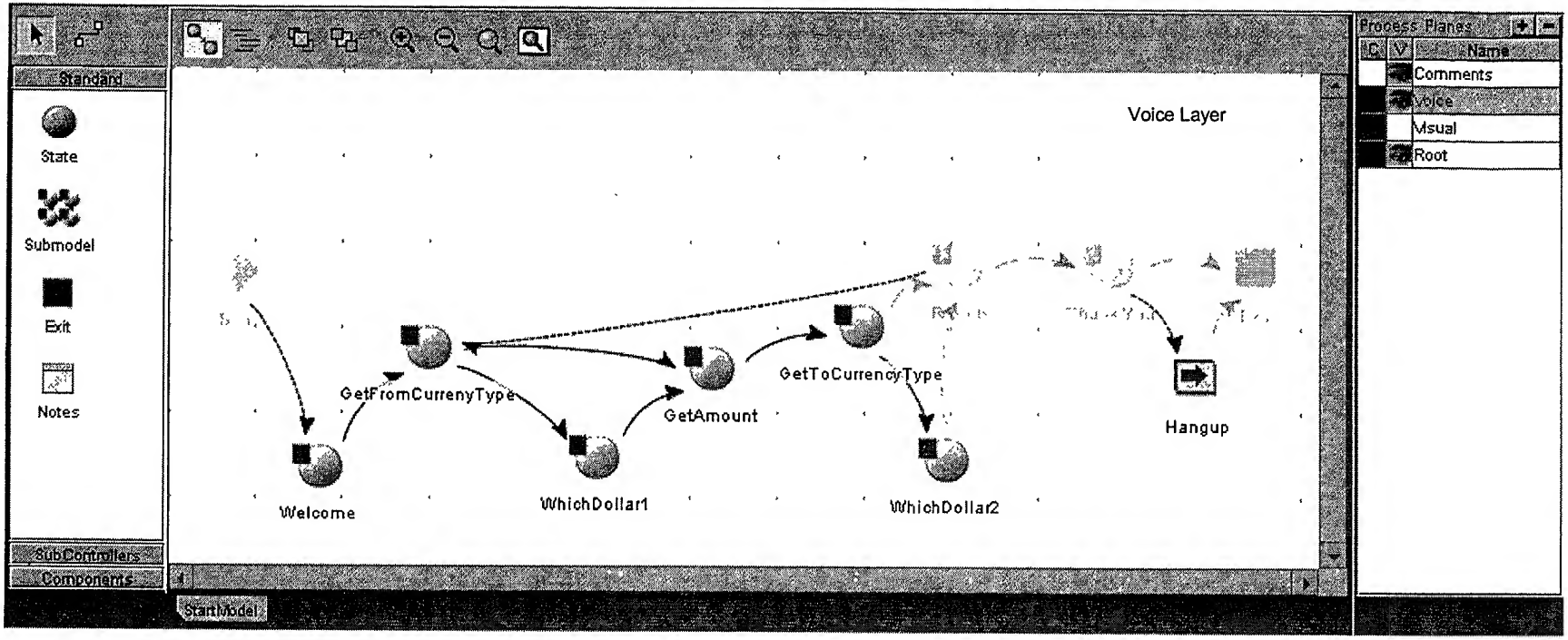
Standard
State
Submodel
Exit
Notes
SubControllers
Components
Voice Layer
GetFromCurrenyType
GetAmount
GetToCurrencyType
Welcome
WhichDollar1
WhichDollar2
Hangup
StartModel
Process Planes
Name
Comments
Voice
Visual
Root

Figure 97

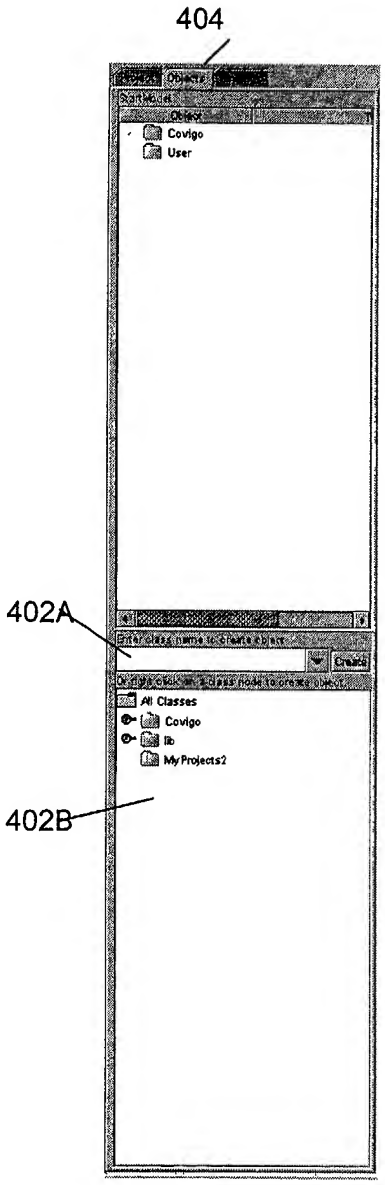
404
402A
402B
Covigo
User
All Classes
Covigo
lib
MyProjects2

Figure 98

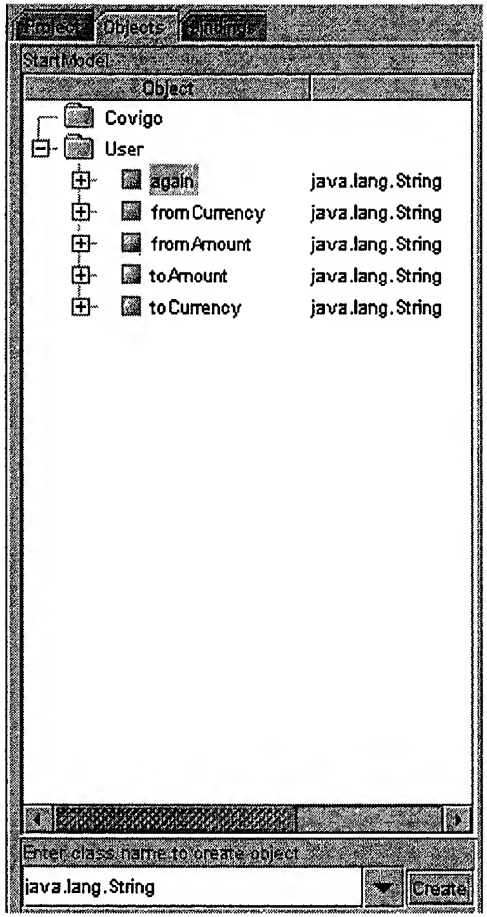
Objects
Start Model...
Object
Covigo
User
again
java.lang.String
fromCurrency
java.lang.String
fromAmount
java.lang.String
toAmount
java.lang.String
toCurrency
java.lang.String
Enter class name to create object
java.lang.String
Create

Figure 99

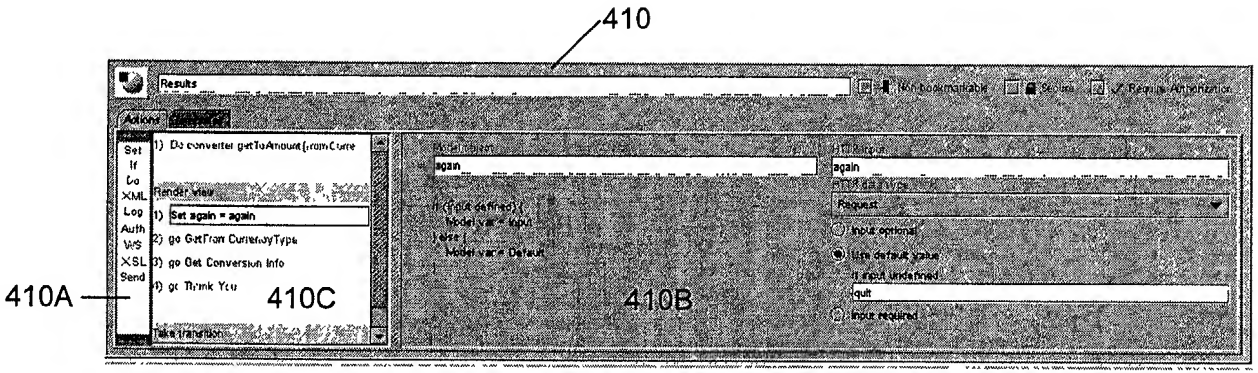

Figure 100

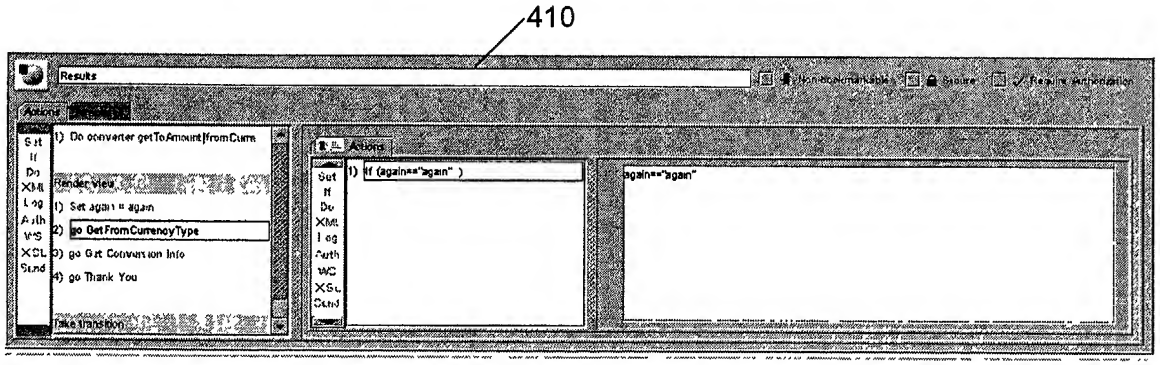

Figure 101

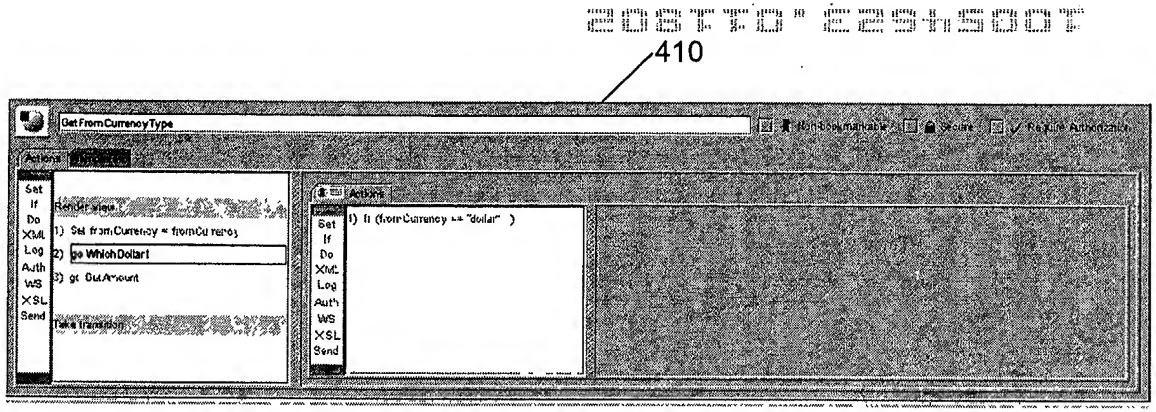
410
GetFromCurrencyType
Actions
Set
If
Do
XML
Log
Auth
WS
XSL
Send
Render view
1) Set fromCurrency = fromCurrency
2) go WhichDollar!
3) gt OutAmount
Take transition
Actions
Set
If
Do
XML
Log
Auth
WS
XSL
Send
1) If (fromCurrency == "dollar")

Figure 102

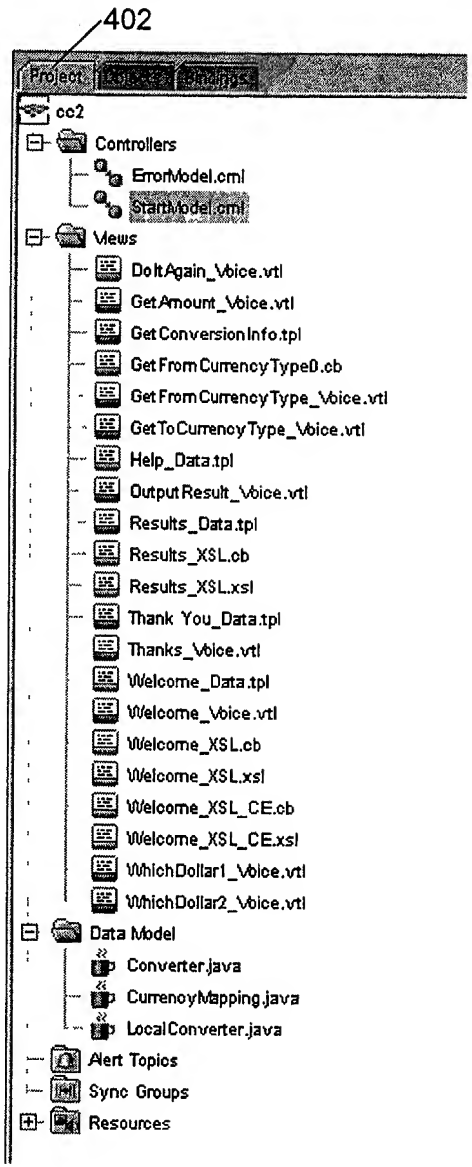
402
Project
cc2
Controllers
ErrorModel.cml
StartModel.cml
Views
DoItAgain_Voice.vtl
GetAmount_Voice.vtl
GetConversionInfo.tpl
GetFromCurrencyType0.cb
GetFromCurrencyType_Voice.vtl
GetToCurrencyType_Voice.vtl
Help_Data.tpl
OutputResult_Voice.vtl
Results_Data.tpl
Results_XSL.cb
Results_XSL.xsl
Thank You_Data.tpl
Thanks_Voice.vtl
Welcome_Data.tpl
Welcome_Voice.vtl
Welcome_XSL.cb
Welcome_XSL.xsl
Welcome_XSL_CE.cb
Welcome_XSL_CE.xsl
WhichDollar1_Voice.vtl
WhichDollar2_Voice.vtl
Data Model
Converter.java
CurrencyMapping.java
LocalConverter.java
Alert Topics
Sync Groups
Resources

Figure 103

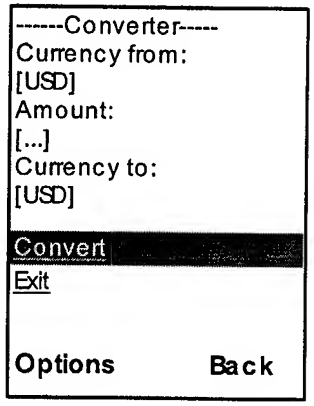

Figure 104

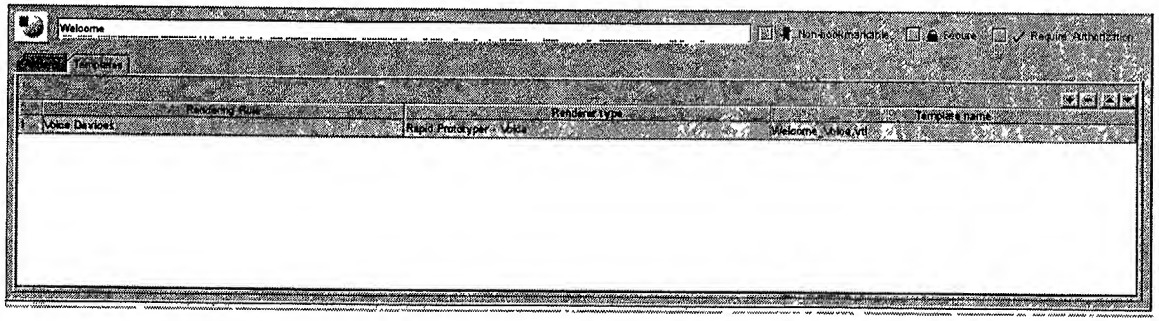

Figure 105

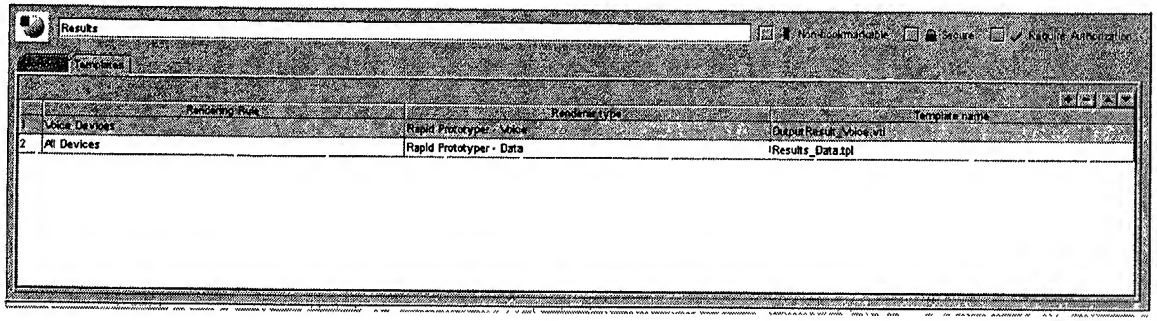

Figure 106

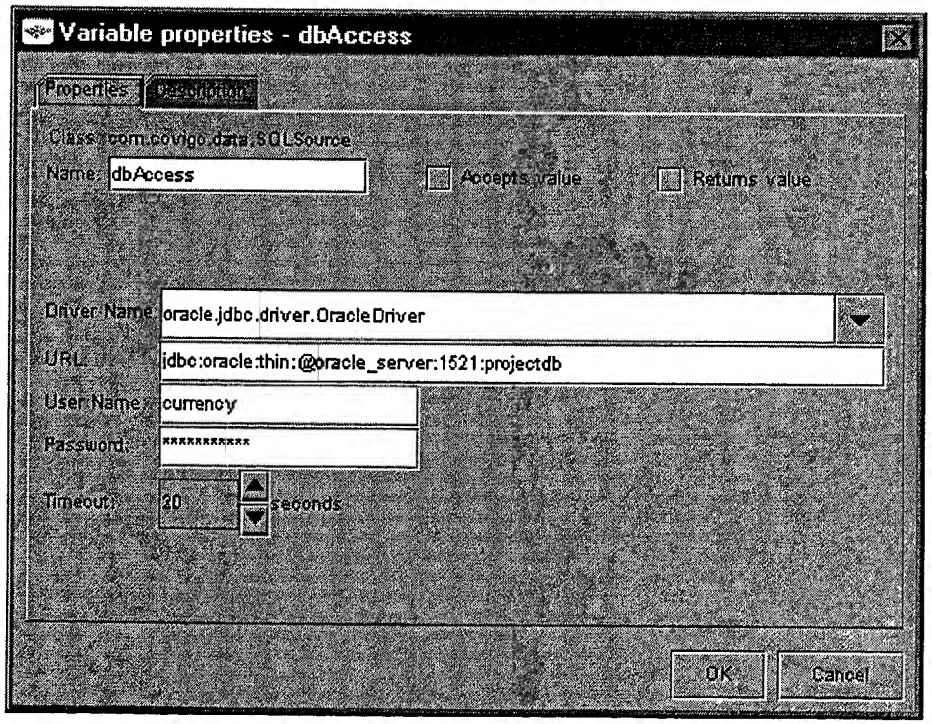

Figure 107

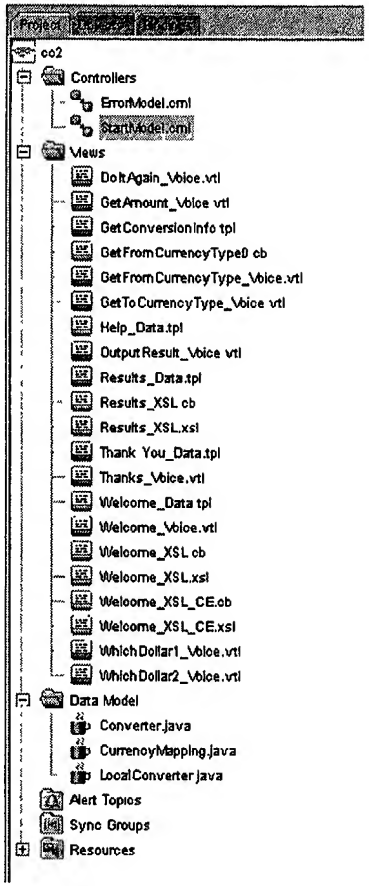
Project
co2
Controllers
ErrorModel.oml
StartModel.oml
Views
DoItAgain_Voice.vtl
GetAmount_Voice.vtl
GetConversionInfo.tpl
GetFromCurrencyType0.cb
GetFromCurrencyType_Voice.vtl
GetToCurrencyType_Voice.vtl
Help_Data.tpl
OutputResult_Voice.vtl
Results_Data.tpl
Results_XSL.cb
Results_XSL.xsl
Thank You_Data.tpl
Thanks_Voice.vtl
Welcome_Data.tpl
Welcome_Voice.vtl
Welcome_XSL.cb
Welcome_XSL.xsl
Welcome_XSL_CE.cb
Welcome_XSL_CE.xsl
WhichDollar1_Voice.vtl
WhichDollar2_Voice.vtl
Data Model
Converter.java
CurrencyMapping.java
LocalConverter.java
Alert Topics
Sync Groups
Resources

Figure 108

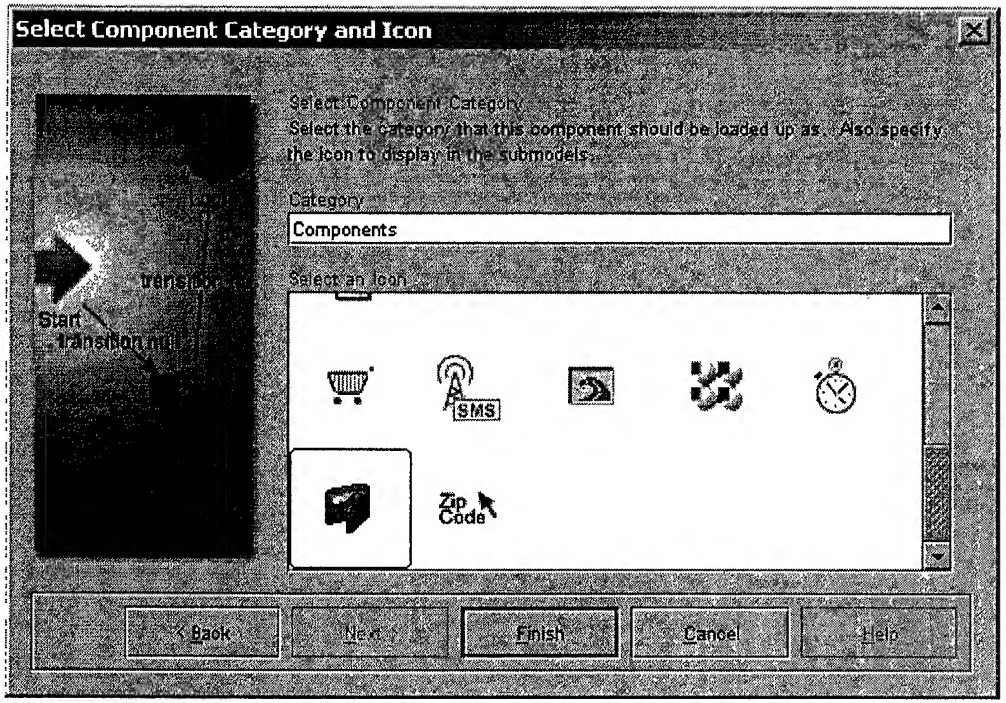
Select Component Category and Icon
Select Component Category
Select the category that this component should be loaded up as. Also specify the icon to display in the submodels.
Category
Components
Select an Icon
transition
Start
transition null
SMS
Zip Code
< Back
Next >
Finish
Cancel
Help

Figure 109

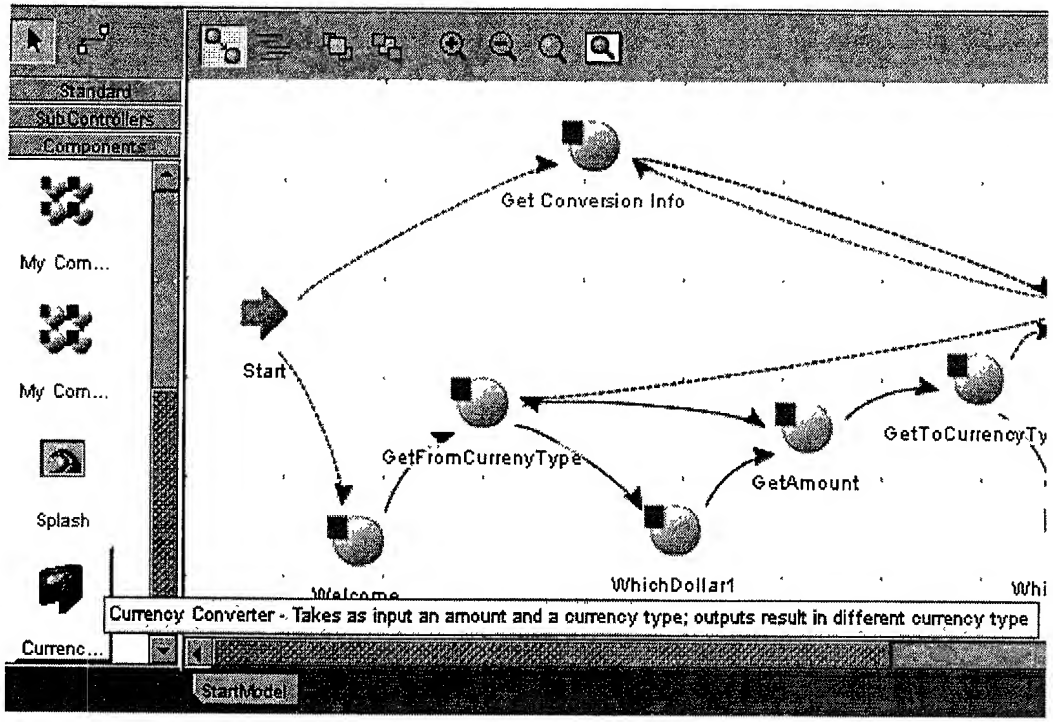
Standard
SubControllers
Components
My Com...
My Com...
Splash
Currenc...
Get Conversion Info
Start
GetFromCurrenyType
GetAmount
GetToCurrencyTy
Welcome
WhichDollar1
Whi
Currency Converter - Takes as input an amount and a currency type; outputs result in different currency type
StartModel

Figure 110

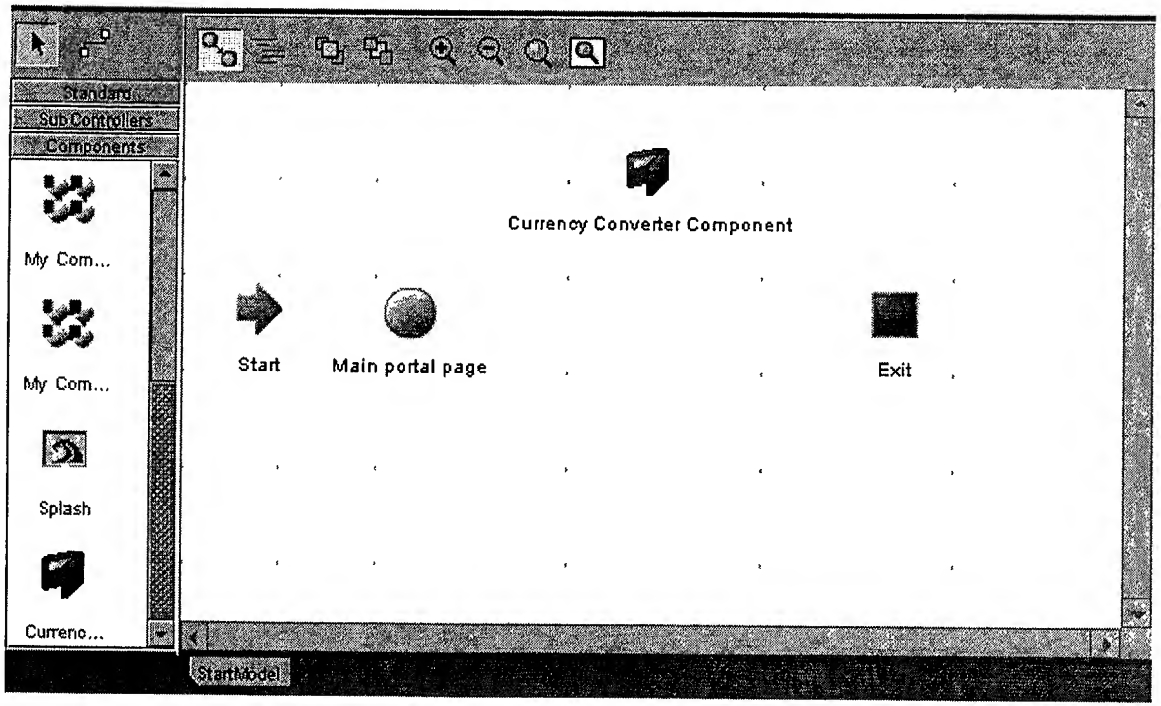
Standard
SubControllers
Components
My Com...
My Com...
Splash
Currenc...
Currency Converter Component
Start
Main portal page
Exit
StartModel

Figure 111

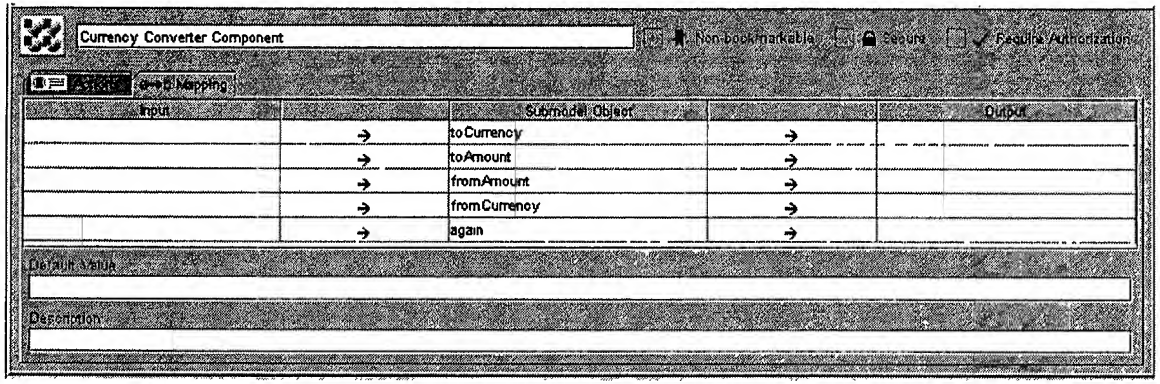
Currency Converter Component
Non-bookmarkable
Secure
Require Authorization
Input
Submodel Object
Output
toCurrency
toAmount
fromAmount
fromCurrency
again
Default Value
Description

Figure 112